Investigation on the digital controller for the BWS

Beam wire scanner meeting - 19.05.2016

J. Emery & P. Andersson

Beams Department

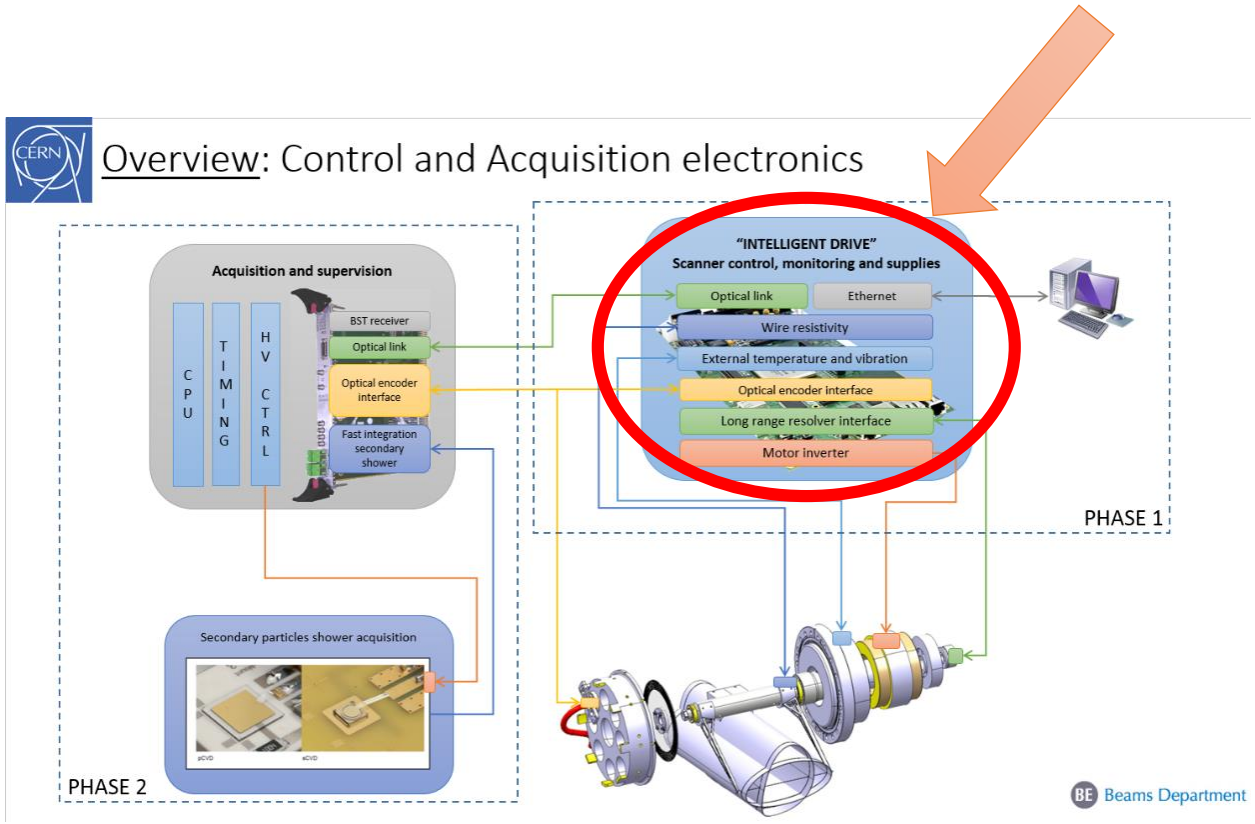
Introduction

- Context of this presentation:
  BI management requested to have technical details on implementation options for the “intelligent drive”

- Goals of this presentation:

1) Provide a sensible overview of options to build final electronics
2) Discuss the pro and cons for each of them
3) List some considerations on FPGA & boards
4) Recommend options based on various constraints

Scope of this presentation



- Focus on the Intelligent drive electronics
- Technical presentation
- Target installation of these drives
- Space in the drive, Power requirement, EMI and measurements
- Options investigated
- FPGA resources, CPU power, design process, DDR
- BWS Firmware architecture and status

Three options for the digital platform

- Option 1:

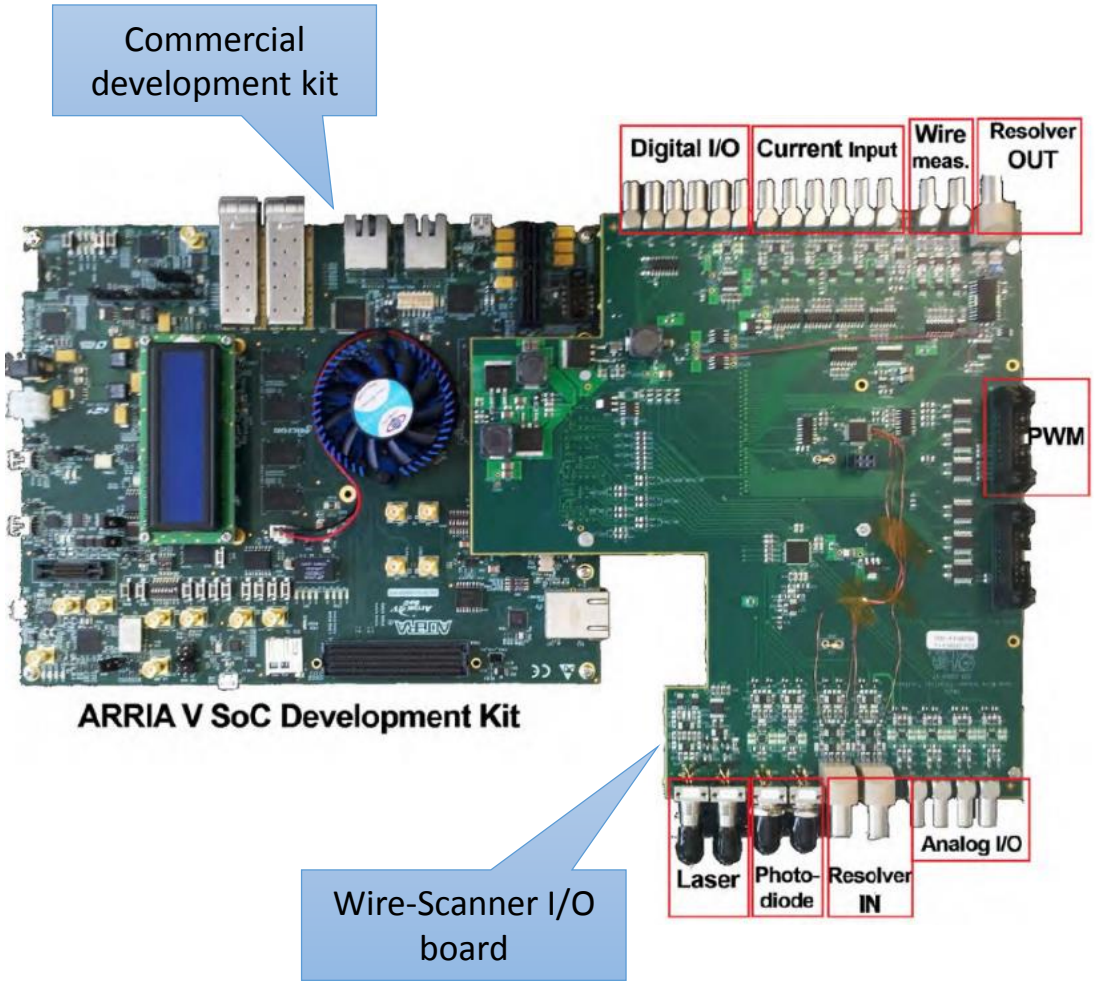
  - Standalone VME board + custom mezzanine
  - Use of the VFC in standalone mode
  - Dedicated analog/optical mezzanine

- Option 2:

  - Combined analogue-digital board
  - Use the FPGA reference design (Altera)
  - Add dedicated analog/optical circuits
  - Combine the 2 boards we have today

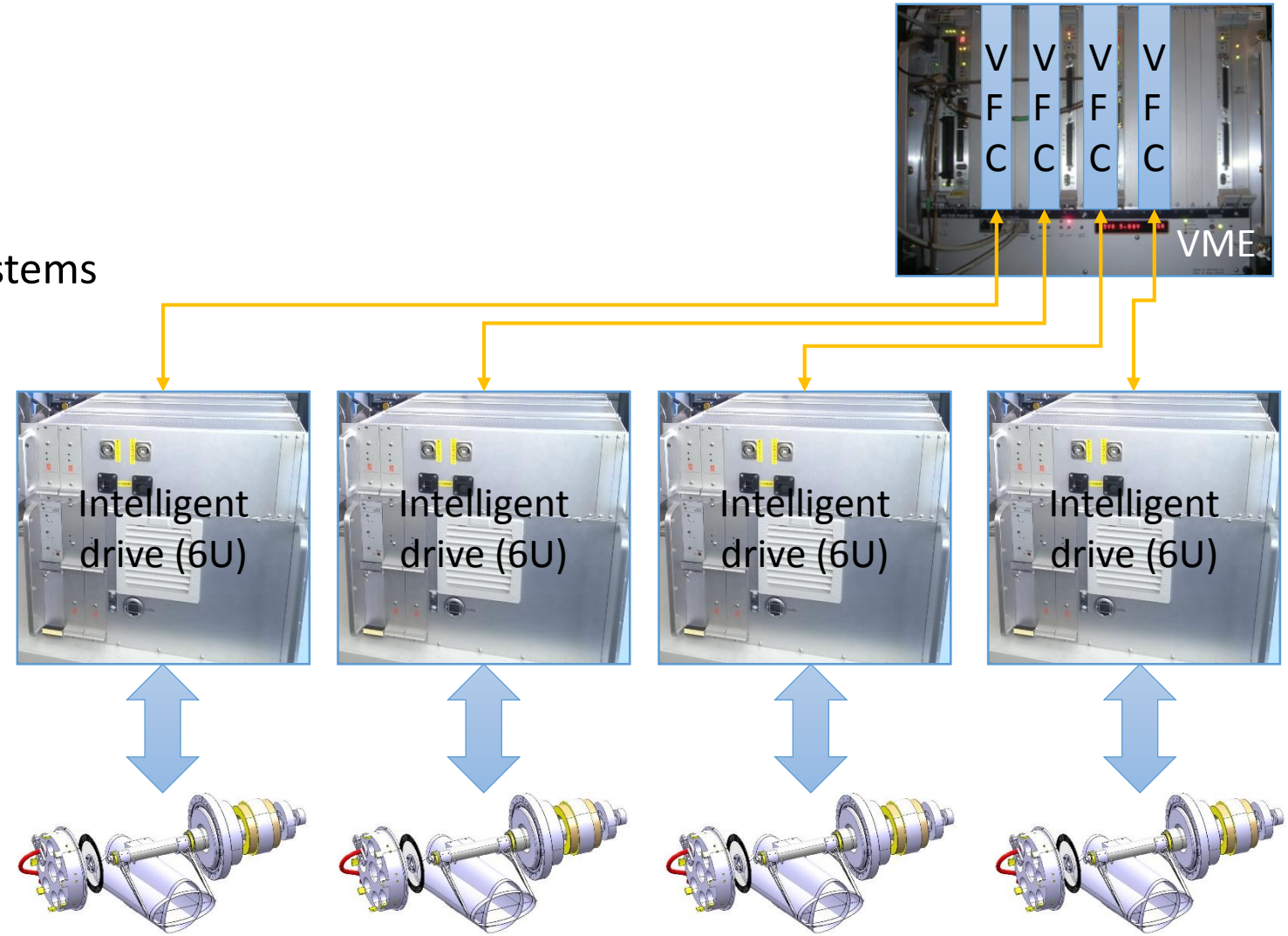
- Option 3: Starter-kit + mezzanine

  - Use Arria V SoC Dev kit
  - Dedicated analog/optical mezzanine

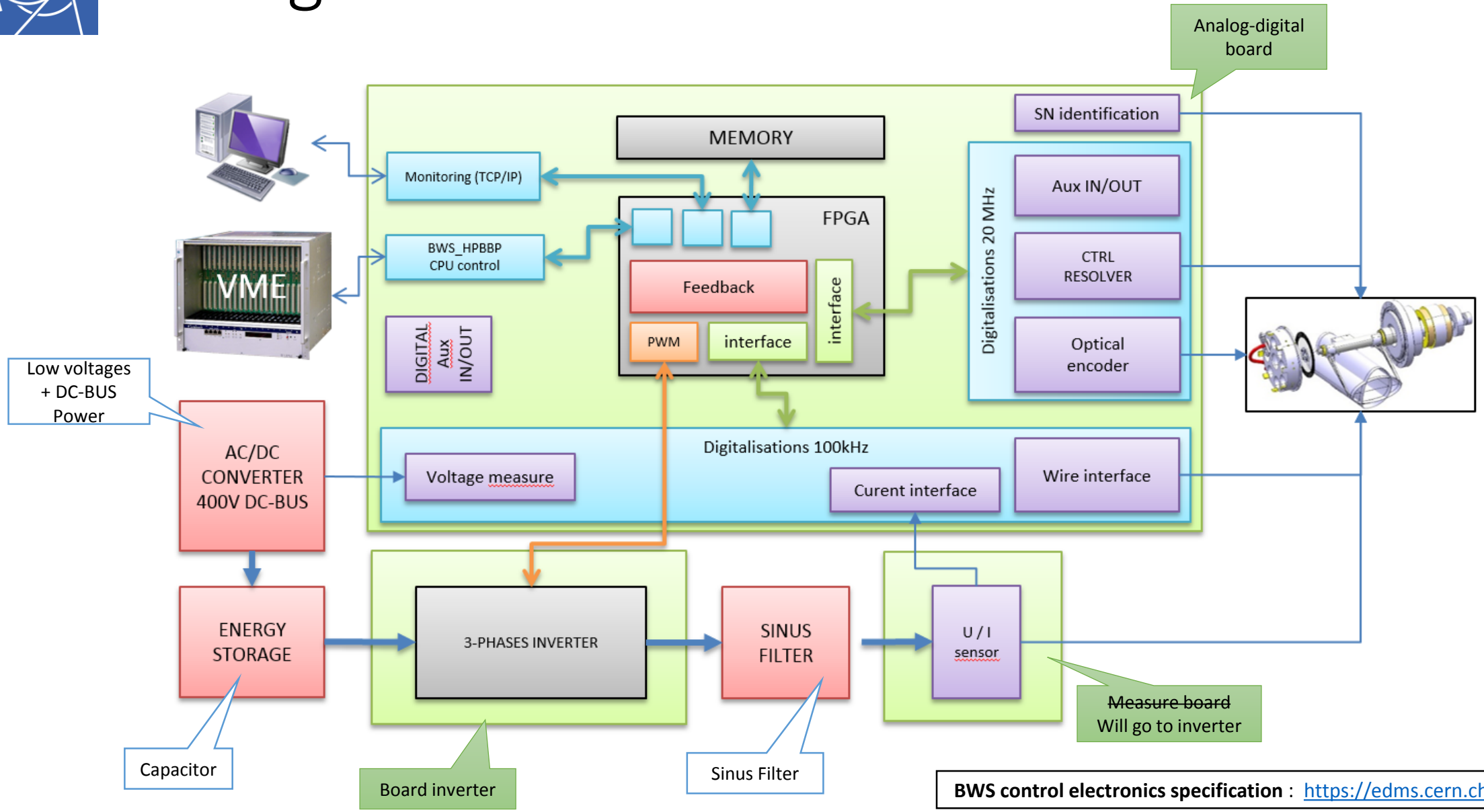


Scanner control architecture

- One intelligent drive (ID) per scanner
  - avoids multiplexing
  - constant monitoring/control
  - allow parallel scans
  - But imply more control and acquisition systems
- Deported processing from VME to ID
- Local monitoring and fault diagnostics
- One VME crate for multiple scanners
  - Number depends on CPU-Memory load
- So we try to minimize the ID size
- Will still require more space than current installation:
  (PSB: 3 racks instead of 1)



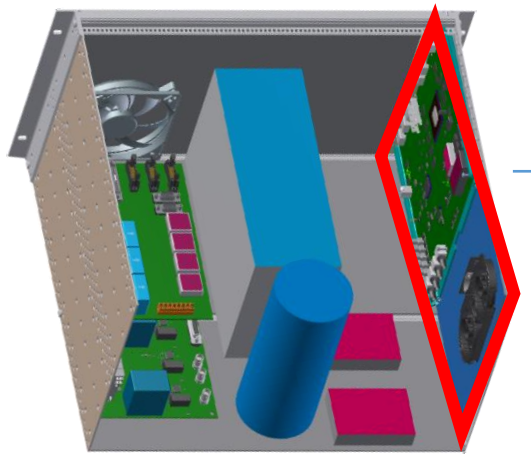
Intelligent drive architecture



BWS control electronics specification : https://edms.cern.ch/document/1318827

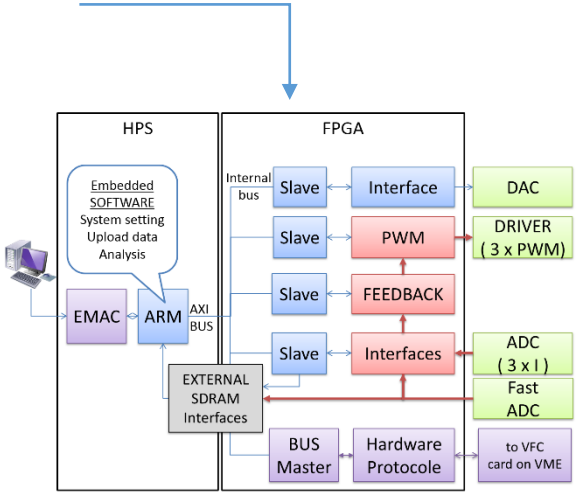
List of selection criteria

- Board realisation
  1) Boards size
  2) Powering requirement
  3) Cabling and EMI protection

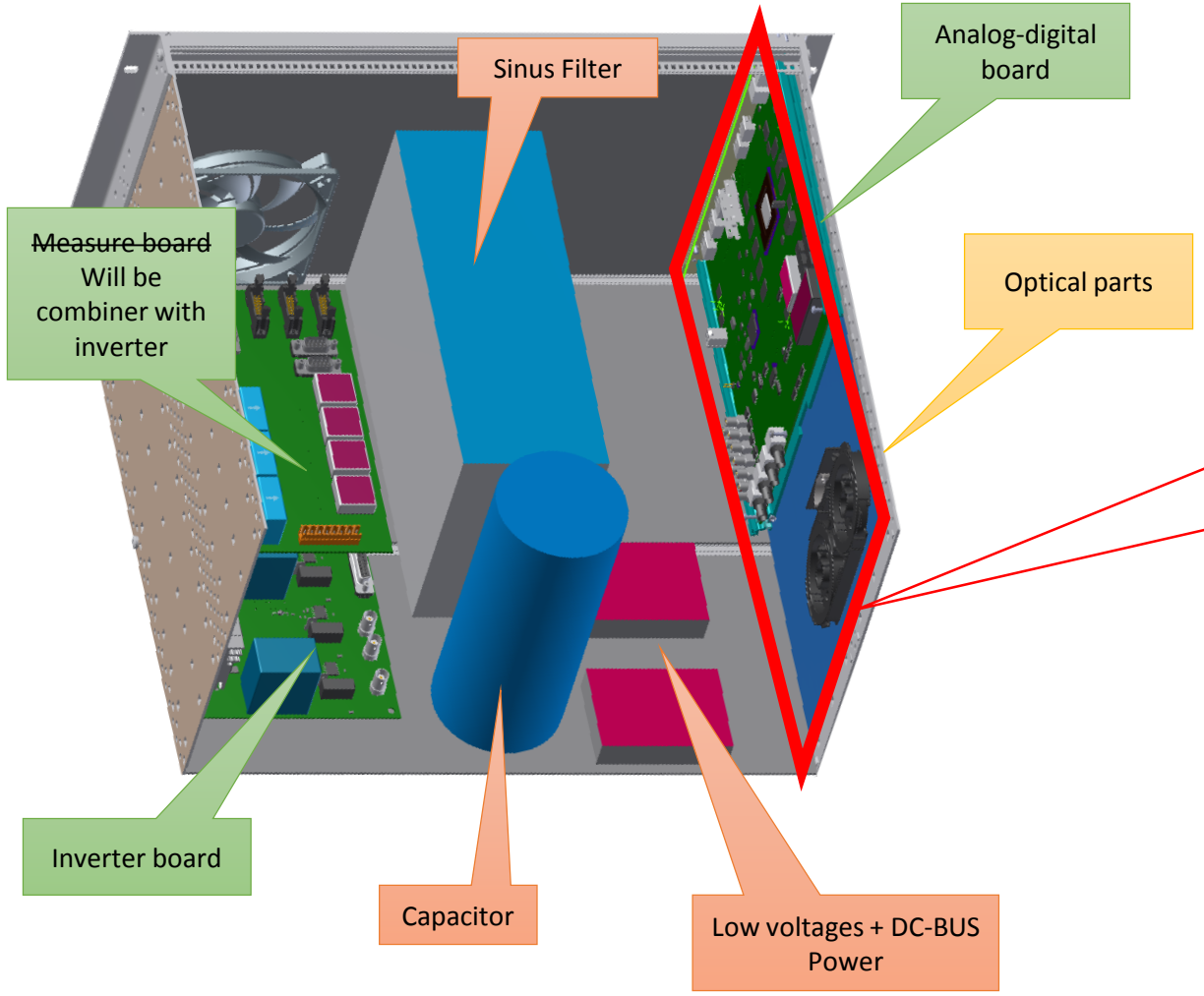


- Digital architecture
  1) FPGA internal resources
  2) FPGA interconnects
  3) External memory

- Design process
  1) Testability
  2) Code reuse
  2) Methodology



Space availability for the board in ID

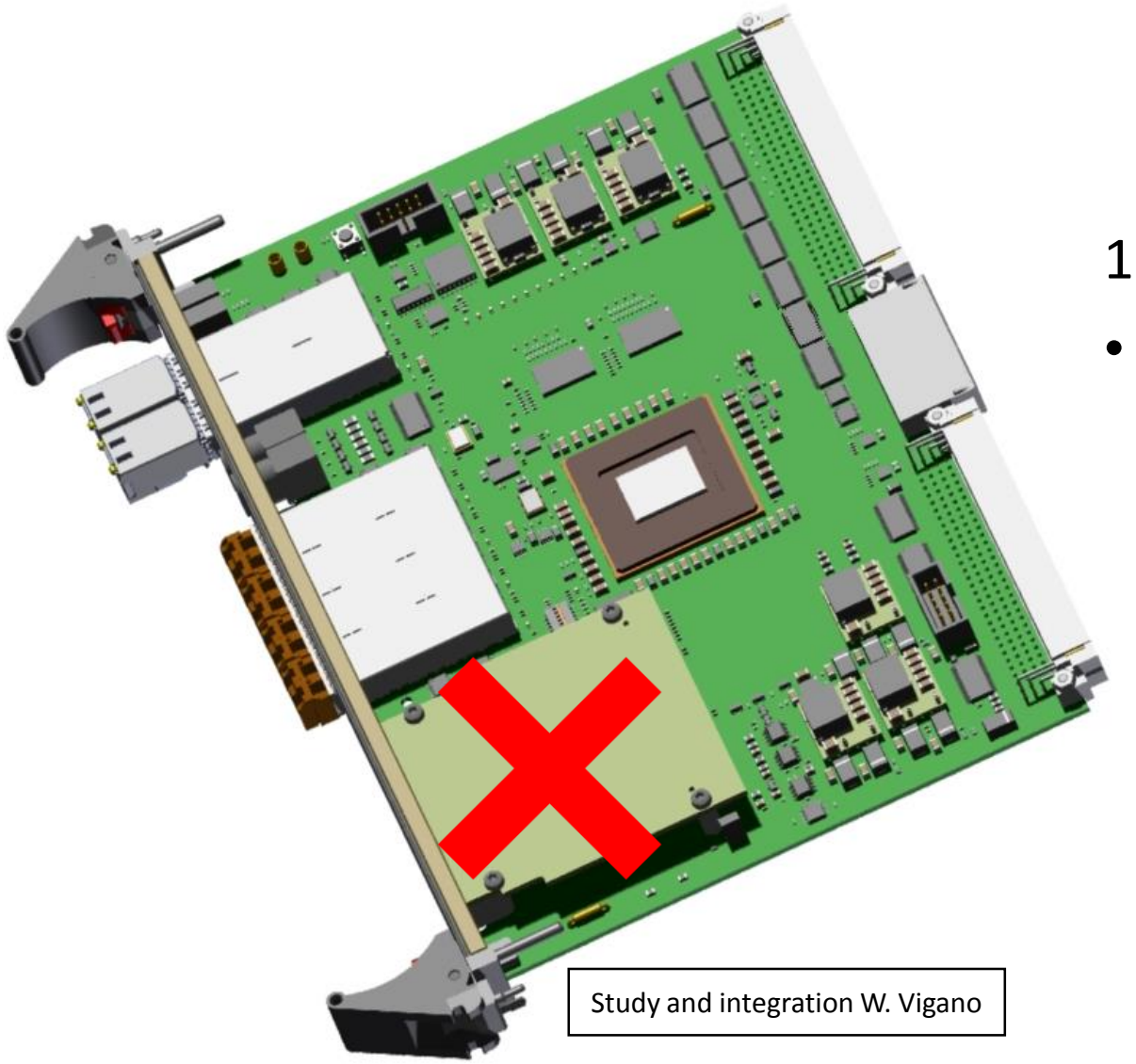


Space available for:
analog, digital and optical

Made possible by reducing:
Inverter size
Power supplies types
internal arrangement
(see Patrik presentation)

Crate study and integration: P. Andersson

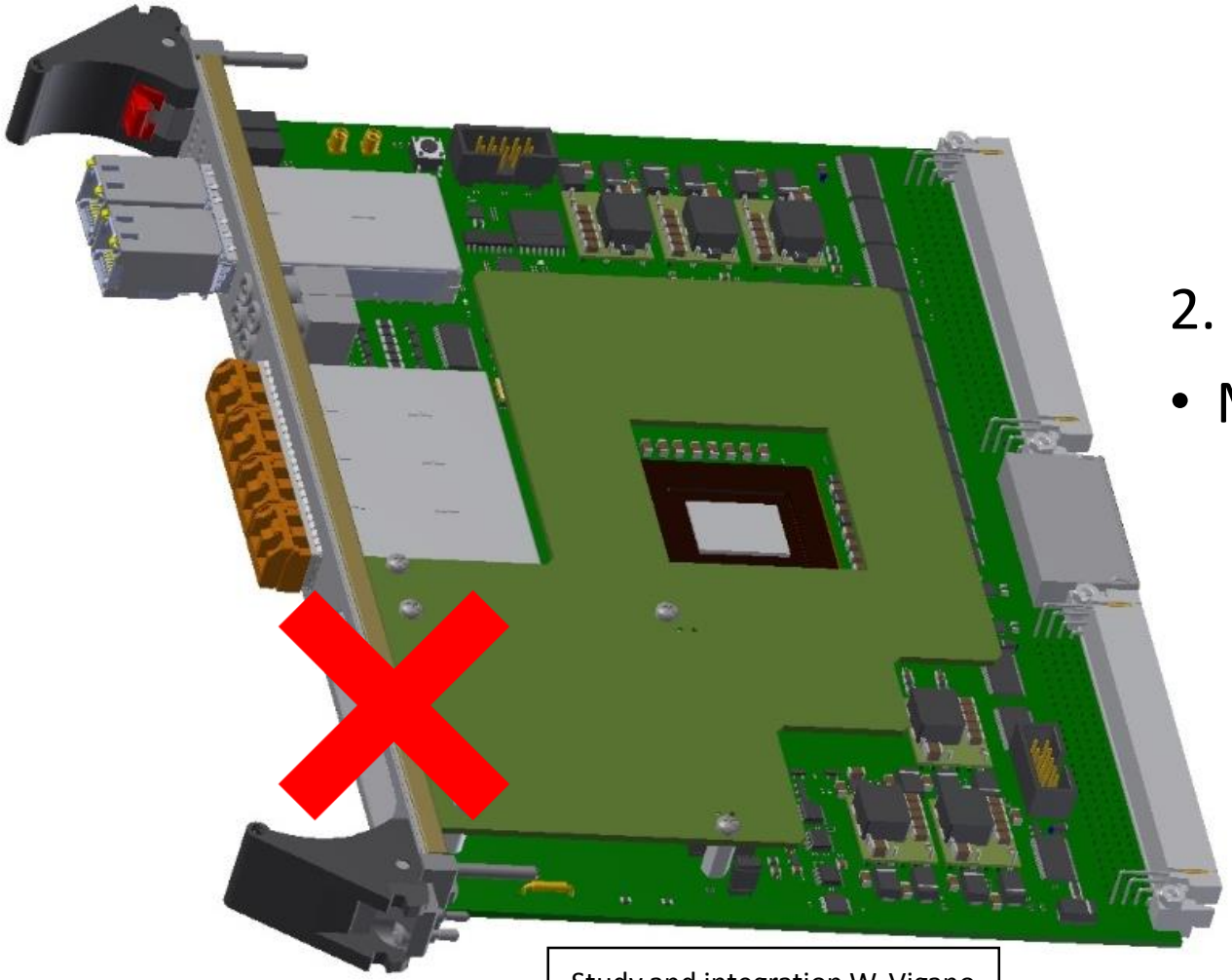
Option 1: VFC + MEZZANINE



Study and integration W. Vigano

1. Standard VFC + standard mezzanine
- Missing board space for components

Option 1: VFC + MEZZANINE



Study and integration W. Vigano

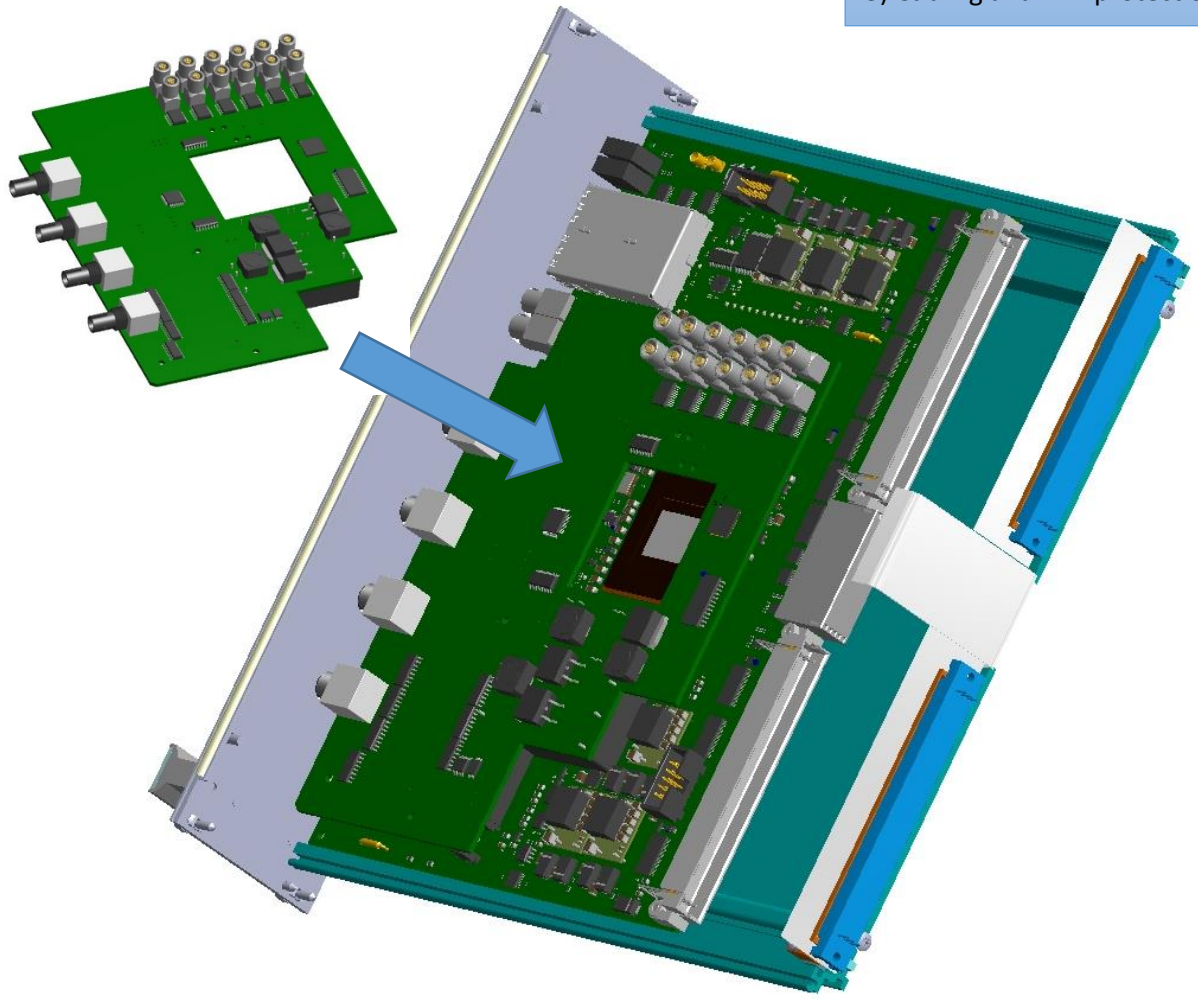
2. Standard VFC + maximized mezzanine

- Missing space for the optical components

Option 1: VFC + MEZZANINE

3. Customized VFC + maximized mezzanine

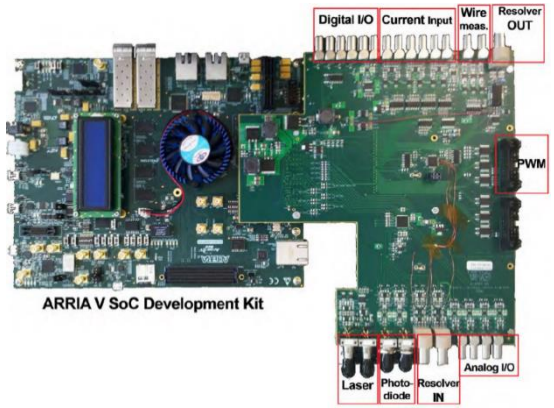
- Remove the lower SPF cages
- Check power supply VADJ
- VFC in standalone mode
- Stack-up of VMC and mezzanine
- Dedicated analogue/optical mezzanine
- None standard FMC mezzanine shape due to the numerous components
- VME connectors for powering the boards

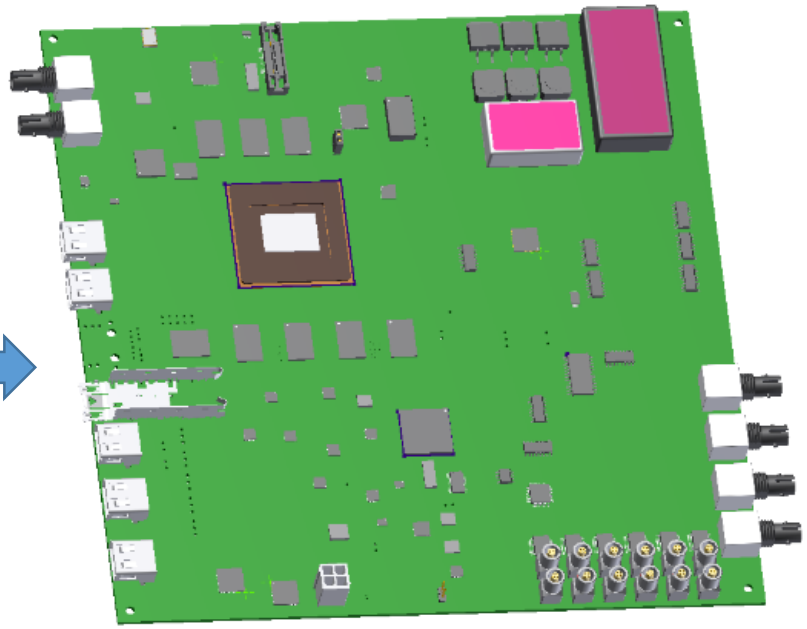


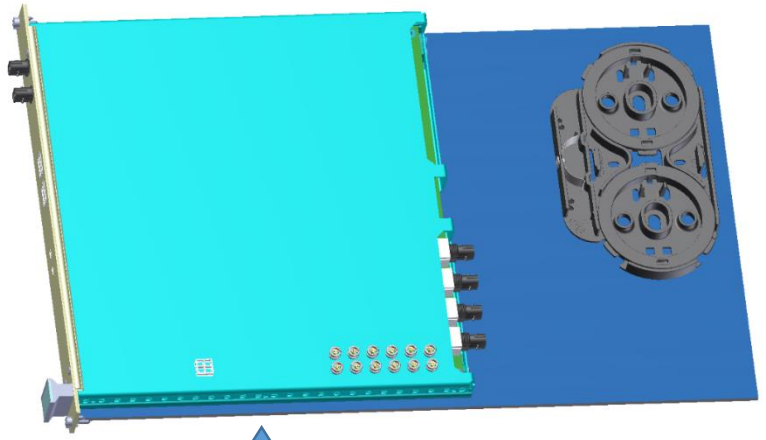
Boards integration and design
W. Vigano & P. Andersson

Option 2: Combined analog-digital board

- Combination of the 2 boards we have today
- Use the FPGA reference design (Altera)



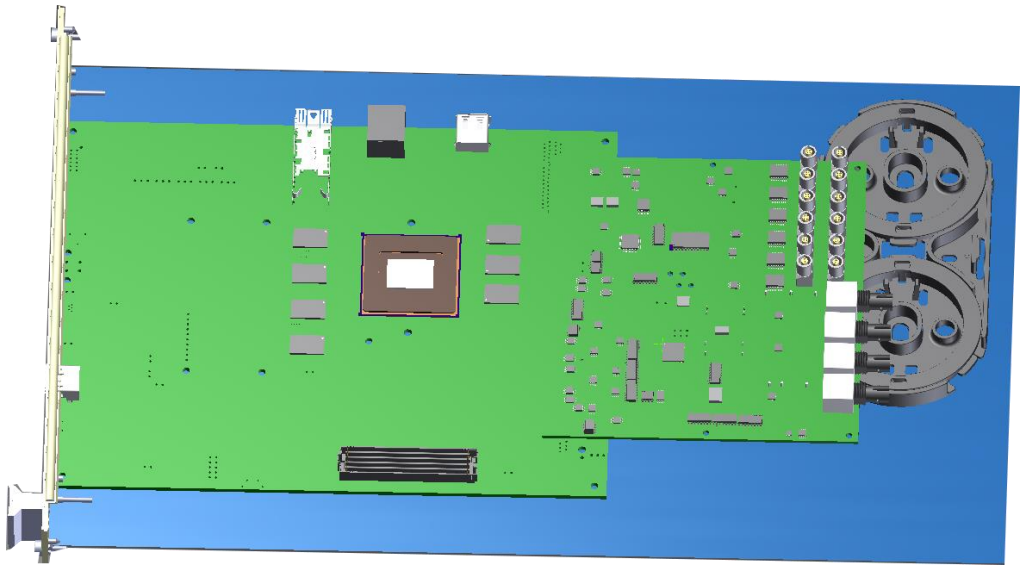


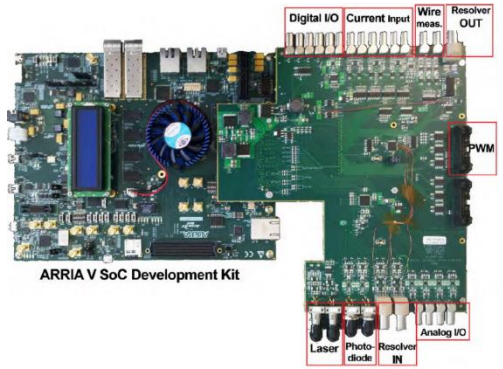


Boards combinations & integration: P. Andersson

Option 3: Starter-kit + mezzanine

- Same configuration as today
- FPGA platform Arria V SoC Dev kit
- Modification of the wire-scanner mezzanine to fit the box



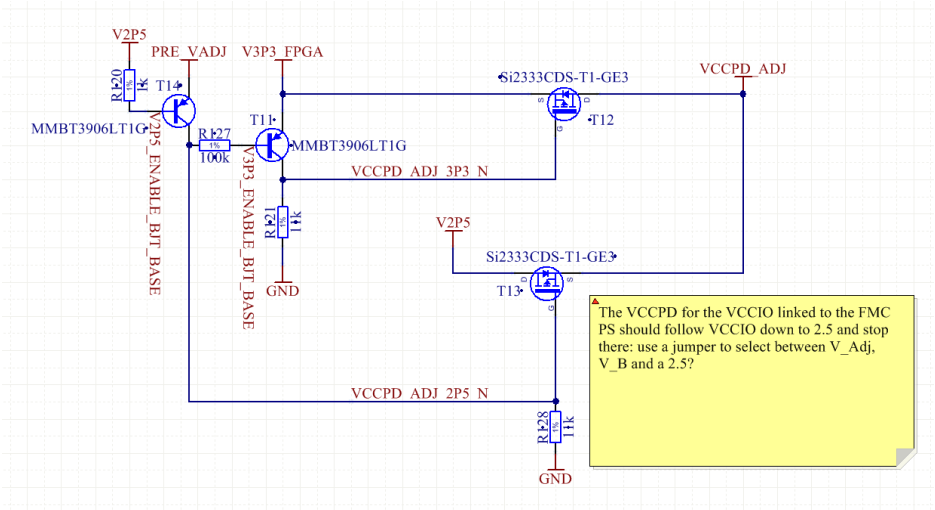


Boards combinations & integration: P. Andersson

Powering requirements

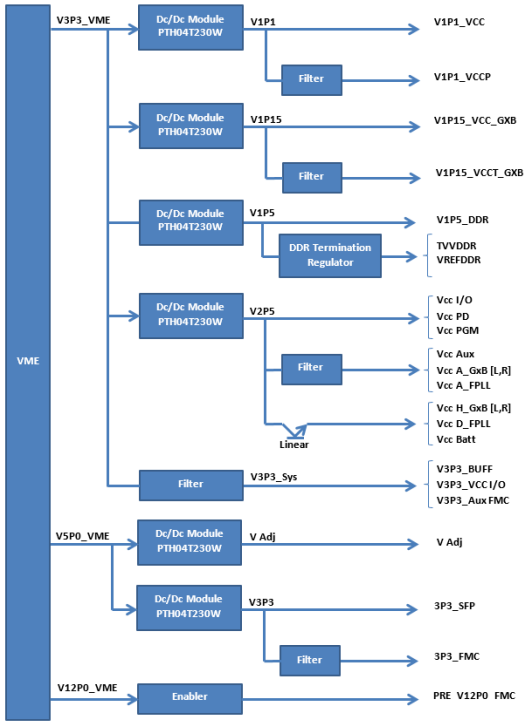
VFC powering the BWS analog board:
12 [V] -> OK
3.3 [V] -> OK
1.8 [V] -> potential problem, limited to 2.5V

1.8[V] is needed to operate the fast ADC, the whole board is using this voltage.

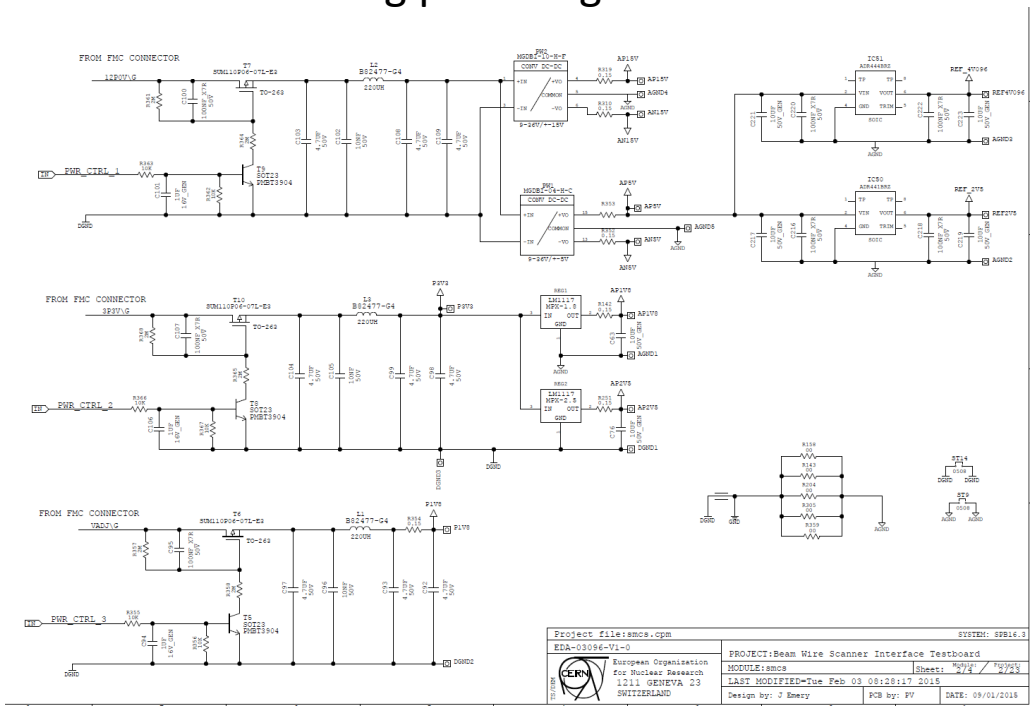


VFC Vadj. Powering schematics

VFC power scheme



BWS analog powering

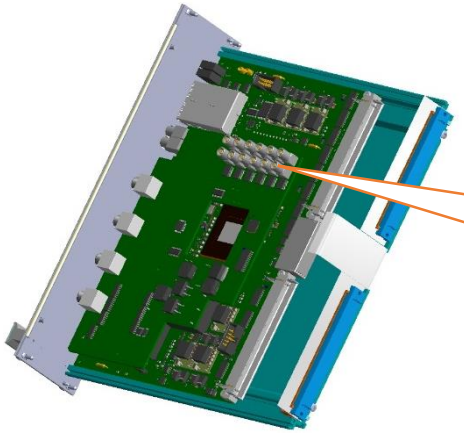


Beams Department

EMI susceptibility

Cablings:

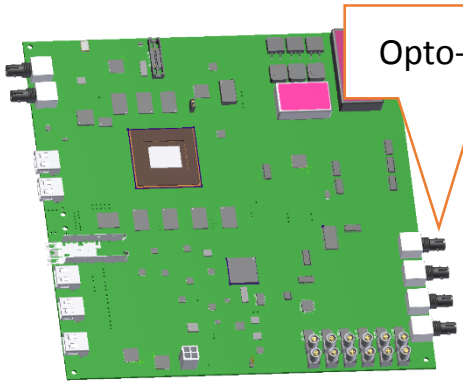
- Similar philosophy of the cablings for all 3 solutions (use of top connectors)
- Other connections slightly worst on the DevKit since uses all sides

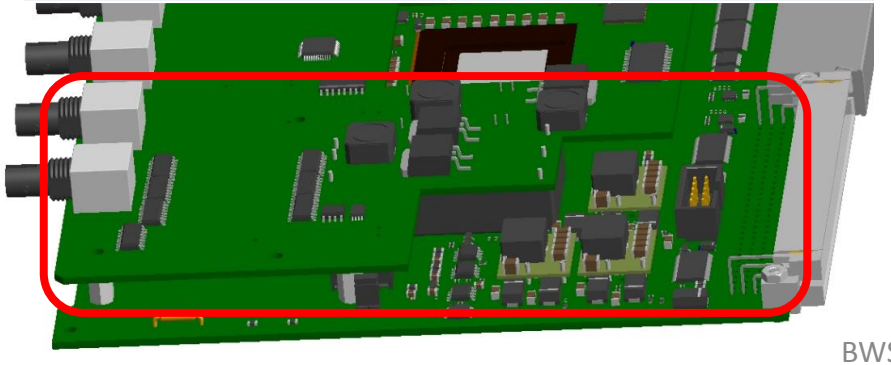


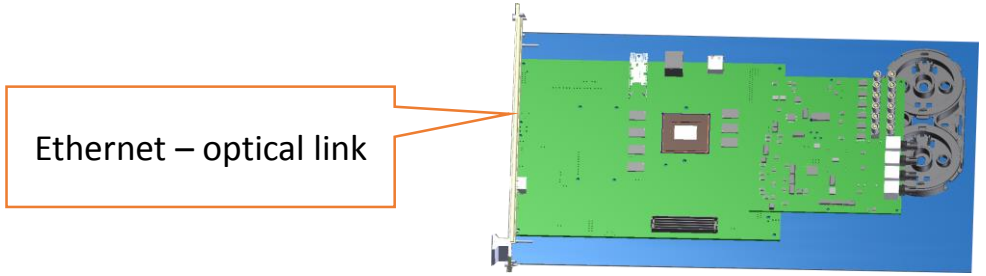
EMI interferences:

- All 3 options will use same box => same shielding from external sources
- Only remains perturbations between analog and digital part

Options	Electrical coupling
1. VFC	-
2. Custom	+
3. DevKit	-







Digital architecture

Digital architecture related criteria:

1) FPGA internal resources
2) FPGA interconnects
3) External memory

Figure 10: Device Chip Overview for Arria V GZ Devices

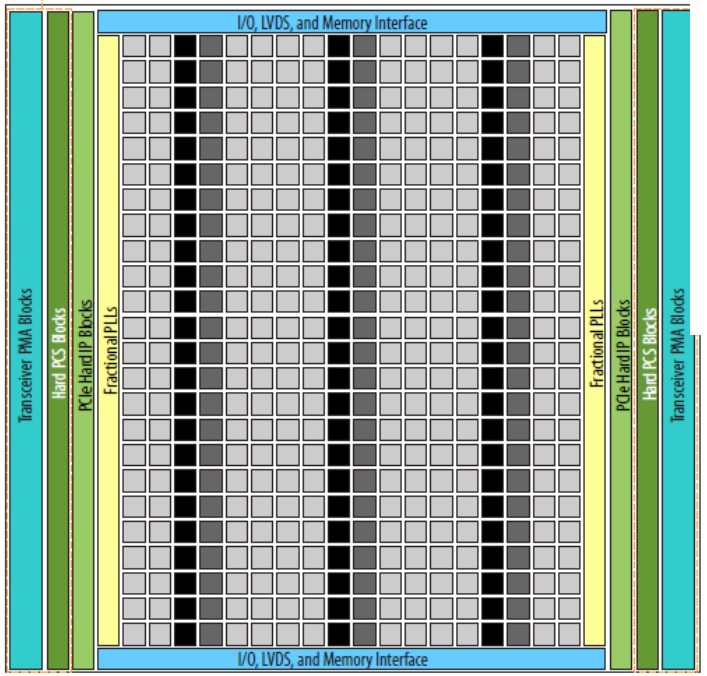
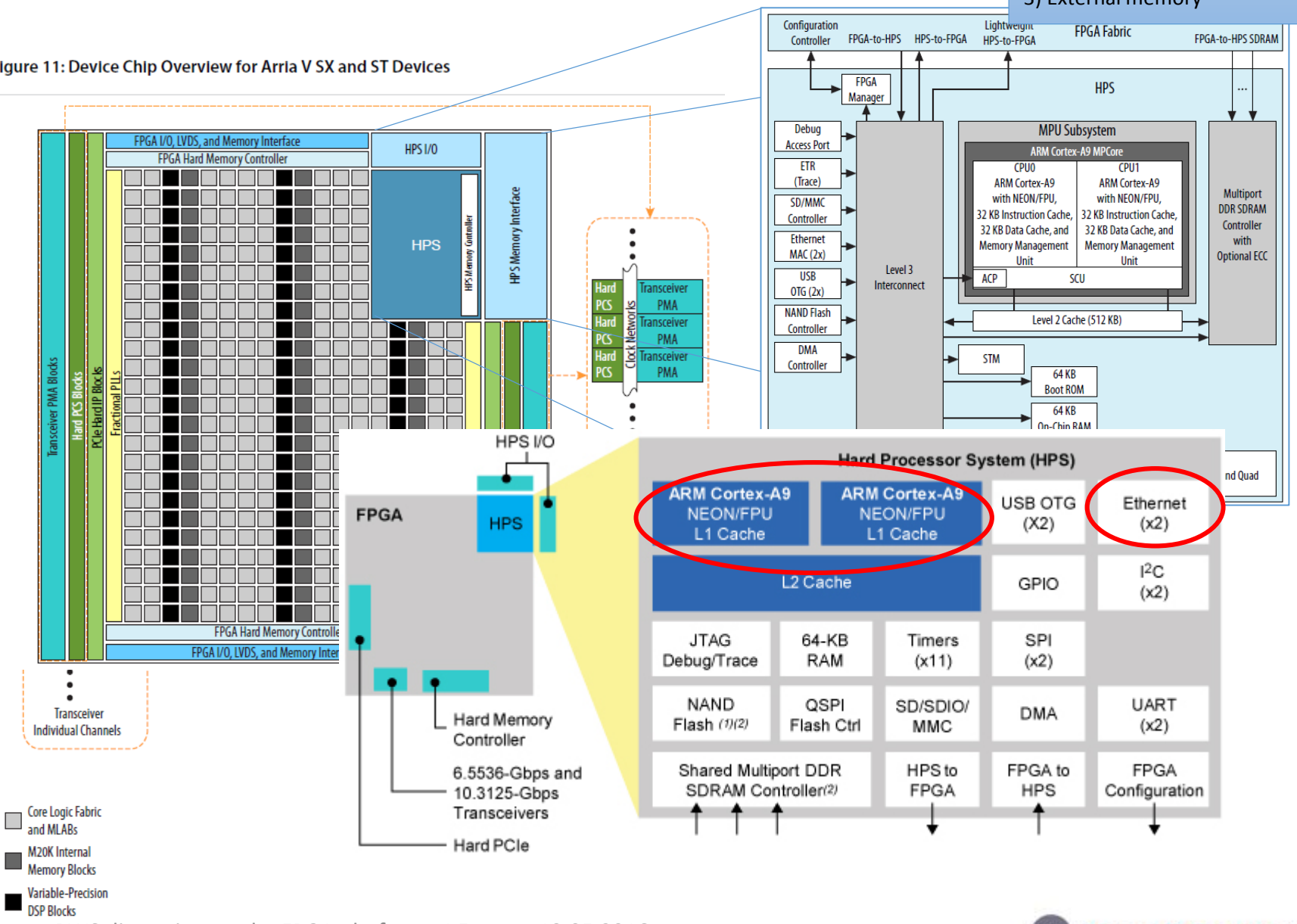


Figure 11: Device Chip Overview for Arria V SX and ST Devices



Digital architecture related criteria

- FPGA logic elements:
  - ok for today's implementation: with the 3 options
  - Future implementation: Depends on processing complexity
- More flexibility using ARM CPU
- External memory potential limitation
- VFC TCP/IP Data transfer will probably be 4 to 10x slower than today (400 Mbits, 100 to 40 Mbits)

	Custom design Analog-Digital	VFC
Code status (for 2016)	95%	50% (6 months)
Evolutivity		Neutral
Use of IP developped for VFC	Same FPGA & transceiver	
FPGA use as today ALM [%]	14	21
Memory [%]	10	16
DSP Blocks [%]	13	16
FPGA type	ARRIA V - SOC	ARRIA V
Type	5ASXFB3H4F40C5N*	5AGXMB1G4F40C4N
Nbr Gates	362K	300K
ALM (adaptive logic module)	136880	113208
Memory (M10k)	17,260	15100
DSP Blocks	1045	920
	2x ARM processor at 1GHz	Software NIOS II at 200 MHz**
Logic use for soft CPUs		4753
Transfert TCP/IP (as today)	>400 Mbits tested point-to-point	40 -100 Mbits max
memory controller	3 hard memory controllers	2 hard memory controllers
Processor side	2x 256 x 16bits + 1 x 256 x 16bits ECC	-
DDR3 SDRAM type	MT41K256M16HA-125:E	MT41K512M16HA-125:E
Organisation	256 M x 16	512 M x 16
FPGA side	4x 256M x 16bits	2x 512M x 16 bits
memory total in bytes	2048 Mbytes	2048 Mbytes
shared with program RAM	no	yes
Nbr of measurement saved (worst case SPS)	2048/336 = 6	2048/336= 6
Maximum theoritical transfert	4 x 1600 Mword = 12.8 Gbyte	2 x 1600 Mword = 6.4 Gbyte
Implemented interface tested	4 x 800 Mword = 6.4 Gbyte/s	Extrapolation: 2x800/4 = 0.8 Gbit/s
SPS: 336 Mbytes burst read time	0.0525	0.105
PSB: 190 Mbyte burst read time	0.0296875	0.059375

With NIOS Softcores

Many parts missing

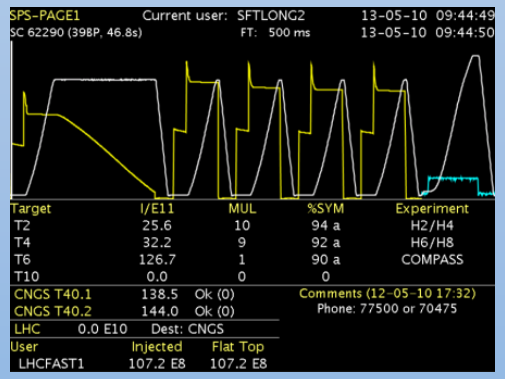
using NIOS Softcores

Details next slides

*As today on the started kit
**Altera "Nios II Performance Benchmarks" 16.12.2015

Overview: External systems connections

General Machine Timing (GMT)
Low jitter < 1ns, Granularity: 1ms



Beam Synchronous timing (BST)
Bunch synchronisation
(25 ns accurate clock)

Revolution frequency synchro
Triggers: scan start, post-mortem
Granularity: 89us (LHC), low jitter < 1ns

Beam Energy and Intensity

CISV?
BCT?

Expert monitoring

FESA class

Settings

Control room (CCC)

Logging storage
Long term storage for offline analysis

C P U

T I M I N G

BST receiver

CISV receiver ?

Optical link

Trigger input

ACQUISITION AND SUPERVISION

Ethernet TCP/IP – RJ45/SFP+

Optical link – SFP+

Start trigger – Digital I/O

LTIM

INTELLIGENT DRIVE

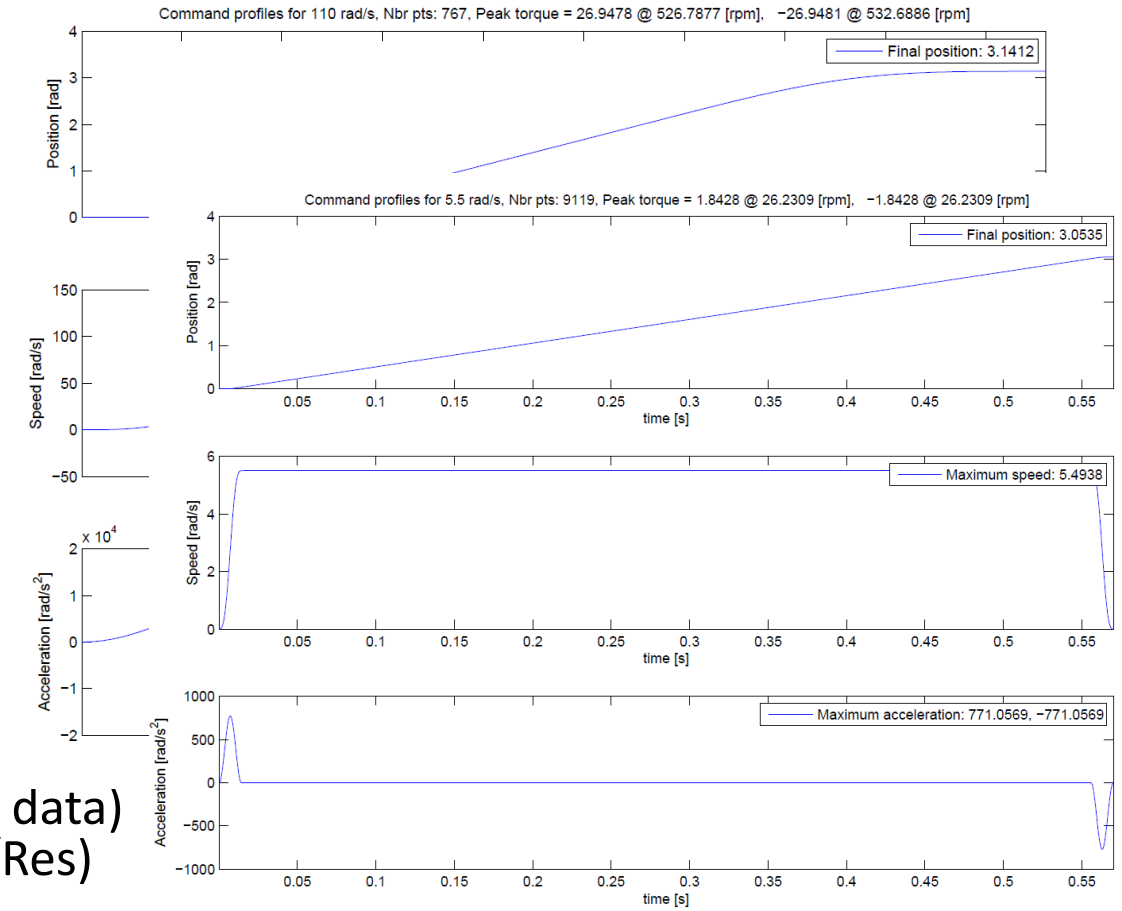
PHASE 1

PHASE 2

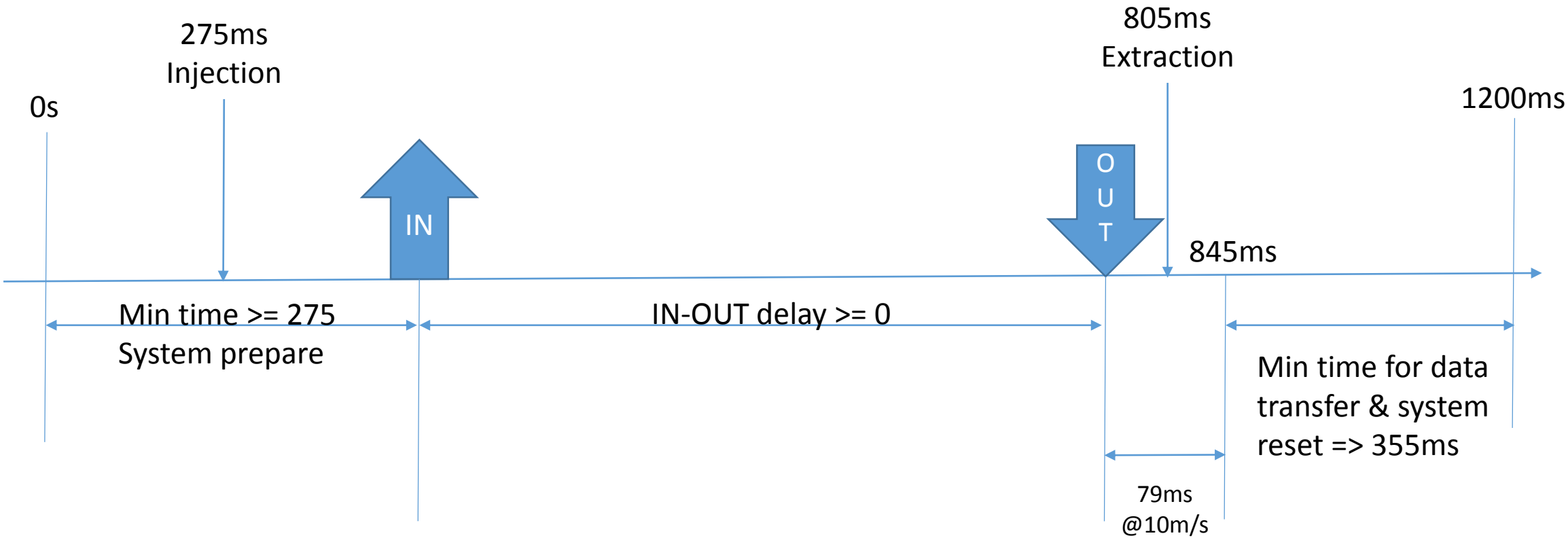
External memory depth requirement

Depends on the use cases

- Scans duration change a lot with speeds
  20 [m/s] -> 48 [ms] (767 pts)
  1 [m/s] -> 570 [ms] (9119 pts)
- Time between IN and OUT
  (can we limit this time to 1s for exemple?)
- Number of scans per user:
  min. 2 if we limit INOUT time to get same functionality
  max. determined by memory depth,
    mode of operation (expert/op),
    required repetition rate (can we fix it?)
- For SPS:
  With time between IN and OUT of 1s
  Worst Case: 2048/336 = 6 scans (full record OPS and resolver data)
  Best Case: 2048/ 5 = 409 scans (no offline processing of OPS/Res)
- VFC or for custom options have the same memory depth (2048 Mbytes)



PSB basic period, 1.2s, one BWS cycle



355ms to transfer 101 Mbytes => 298 Mbit/s
Not possible with TCP/IP and VFC
Needs full implementation using VME (Phase 2)

Memory depth for the PSB

Expert mode (detailed data)

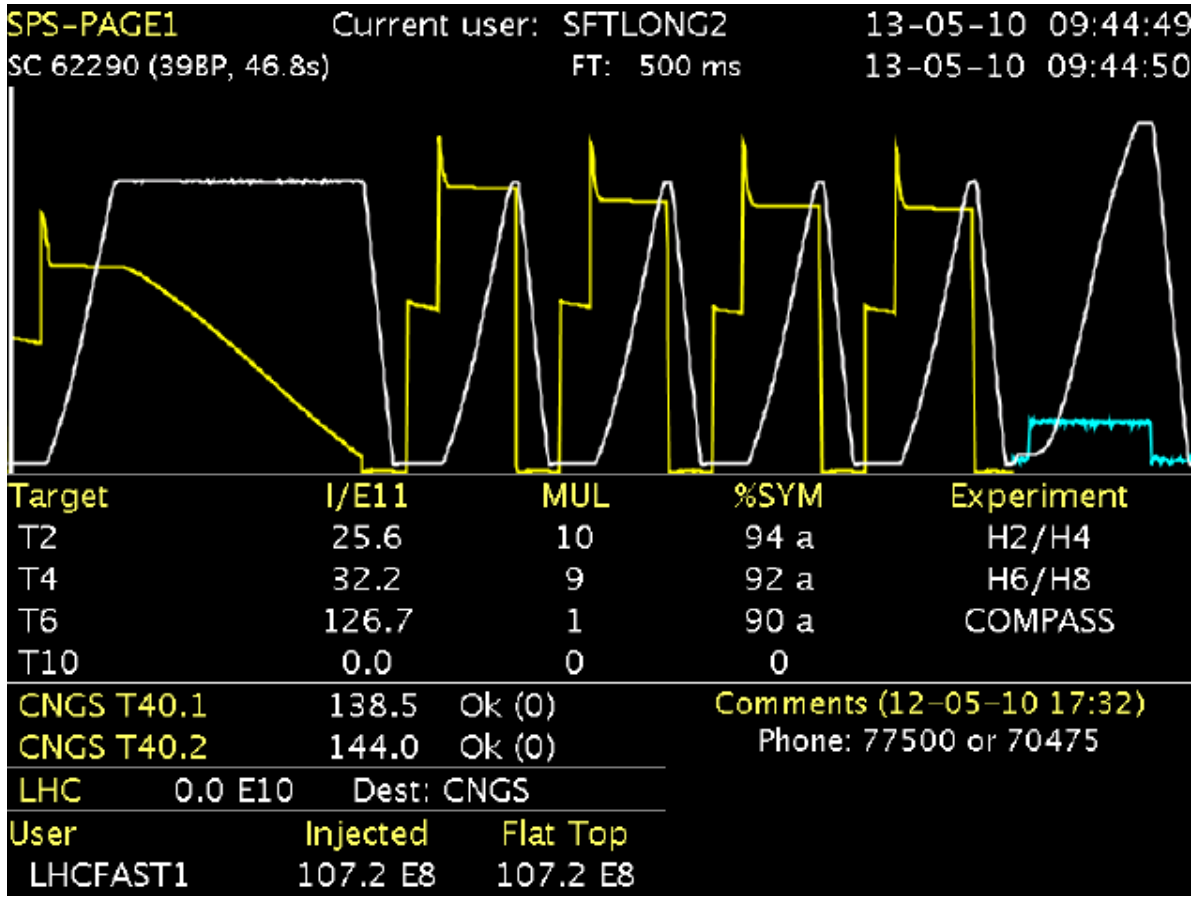
Tangential speed [m/s]	Angular speed (PSB)	movement duration [s]	max. INOUT	feedback + wire data [Mbits]	Optical encoder [Mbits]	resolver raw [Mbits]	total [Mbits]	[Mbyte]
20	133	0.04	0.51	20.33	360.11	360.11	740.54	92.57
15	100	0.053	0.504	21.02	372.31	372.31	765.65	95.71
10	67	0.0785	0.491	22.33	395.51	395.51	813.34	101.67
1	6.7	0.485	0.245	41.86	741.58	741.58	1525.02	190.63
20	133	0.04	0.51	20.33	0.58	0.29	21.20	2.65
15	100	0.053	0.504	21.02	0.60	0.30	21.92	2.74
10	67	0.0785	0.491	22.33	0.64	0.32	23.28	2.91
1	6.7	0.485	0.245	41.86	1.20	0.60	43.66	5.46

Operational mode (OPS & RESOLVER processed)

1) One measurement cycle (one IN, one OUT)
2) Data recorded continuously between in and out movement
3) INOUT time calculated for IN=275ms, OUT=805ms
4) Expert mode => Motion data and raw encoders data storage
   Will be used until we have the optical position sensor digitalised in the VME
5) Op Mode => Motion data and processed encoders data storage

IN-OUT delay used for the calculation:
20 m/s => 805-20-275=510ms
15m/s => 805-53/2-275=504ms
10m/s => 805-79/2-275=491ms
1m/s => 805-570/2-275=245ms

SPS Supercycle
multiple BWS cycles per user



- Multiple users with different durations 1.2s to >20s
- Challenge arises when managing multiple scanning cycle per user
- Clear use cases must be given by OP/ABP to calculate the among of produced data and what data reduction we need to apply:
  - at the FPGA levels (ID and/or AS)
  - at the VME - CPU levels

Memory depth for the SPS

Digital architecture
1) FPGA internal resources
2) Board interconnects
3) External memory

Expert mode (detailed data)

Tangential speed [m/s]	Angular speed (SPS)	movement duration [s]	INOUT	feedback + wire data [Mbits]	Optical encoder [Mbits]	resolver raw [Mbits]	total [Mbits]	[Mbyte]
20	110	0.048	1	37.76	668.95	668.95	1375.65	171.96
15	82	0.064	1	38.87	688.48	688.48	1415.82	176.98
10	55	0.0933	1	40.88	724.24	724.24	1489.37	186.17
1	5.5	0.57	1	73.73	1306.15	1306.15	2686.04	335.75
20	110	0.048	1	37.76	1.08	0.54	39.38	4.92
15	82	0.064	1	38.87	1.11	0.56	40.53	5.07
10	55	0.0933	1	40.88	1.17	0.58	42.64	5.33
1	5.5	0.57	1	73.73	2.11	1.05	76.89	9.61

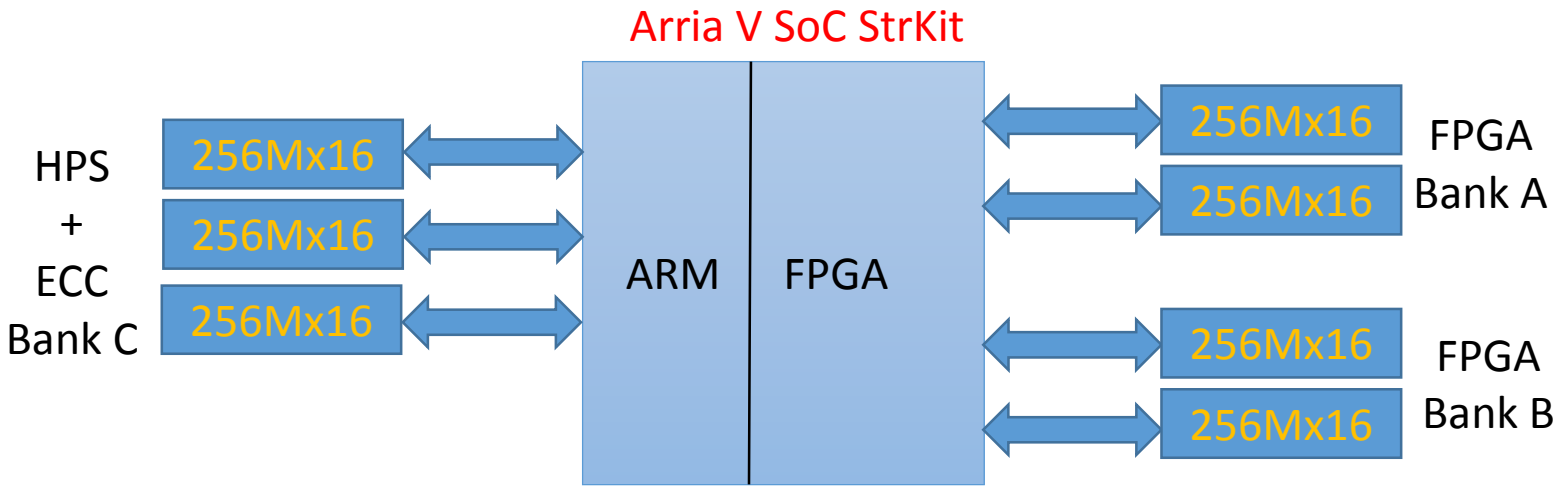
Tangential speed [m/s]	Angular speed (SPS)	movement duration [s]	INOUT	feedback + wire data [Mbits]	Optical encoder [Mbits]	resolver raw [Mbits]	total [Mbits]	[Mbyte]
20	110	0.048	10	347.86	6162.11	6162.11	12672.08	1584.01
15	82	0.064	10	348.96	6181.64	6181.64	12712.24	1589.03
10	55	0.0933	10	350.98	6217.41	6217.41	12785.80	1598.22
1	5.5	0.57	10	383.83	6799.32	6799.32	13982.47	1747.81
20	110	0.048	10	347.86	9.94	4.97	362.77	45.35
15	82	0.064	10	348.96	9.97	4.99	363.92	45.49
10	55	0.0933	10	350.98	10.03	5.01	366.02	45.75
1	5.5	0.57	10	383.83	10.97	5.48	400.28	50.04

Operational mode (OPS & RESOLVER processed)

1) Only one measurement cycle (one IN, one OUT)
2) Data recorded continuously between in and out movement
3) Large data grow due to INOUT delay => <u>Can we limit INOUT to a maximum of 1s and do multi-scans per cycle?</u>
4) Expert mode => Motion data and raw encoders data storage
   <u>Will be used until we have the optical position sensor digitalised in the VME</u>
5) Op Mode => Motion data and processed encoders data storage

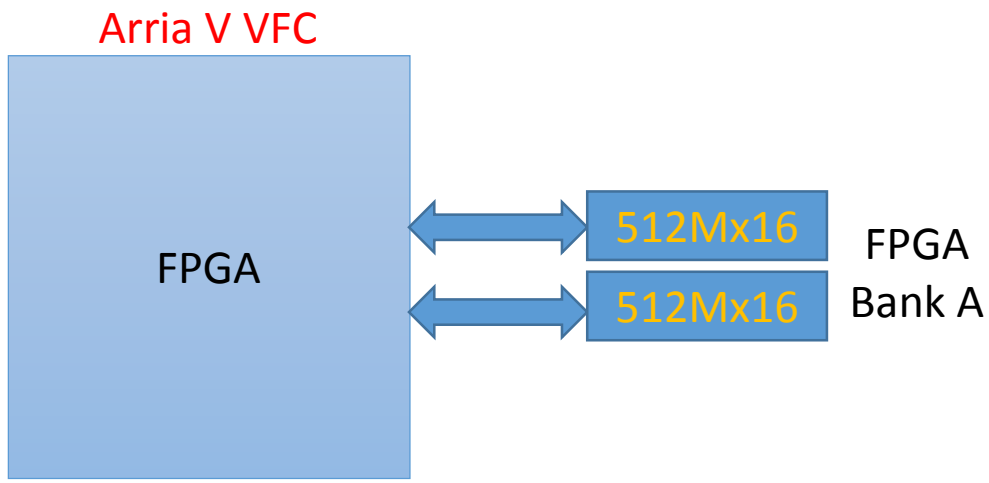
External memory access organisation

Is the external memories connections could be a limitation?



Theoretical transfer: 12.8 Gbit/s
Implemented: 6.4 Gbit/s

Bank A: Fast ADC => 4x20MHzx16bits => 1.28 Gbit/s
Bank B: Other ADC => 70*16k*32bits => 36.12 Mbits/s
Bank C: ARM dualcore ram memory



Theoretical transfer: 6.4 Gbit/s
Implemented ?: 3.2 Gbit/s

Bank A:
- 4x20MHzx16bits => 1.28 Gbit/s
- 70*16k*32bits => 36.12 Mbits/s
- Softcore (Nios) RAM?

Efficiency reduction due to arbiter + R/W and Random access:
- write to multiple memory location (20M & 16K): ? 25%
3.2 Gbit/s / 4 => 0.8 Gbit/s => Potential limitation for OPS + Resolver

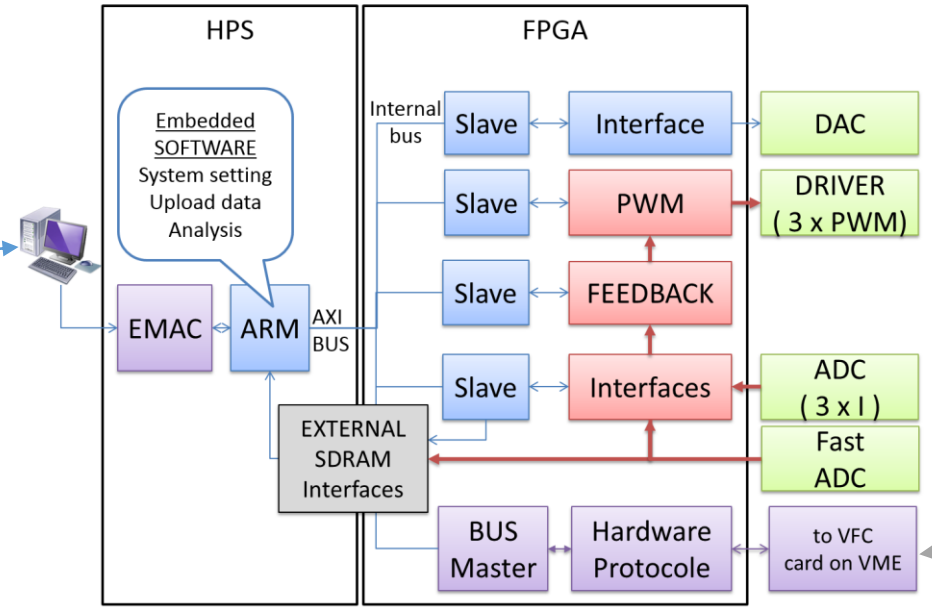
System & Firmware testability

Is the FPGA selection will determine the system and Firmware testability?

- Simulation level:
  Most of the final code written in VHDL => Verification (VHDL, SystemVerilog, Simulink) on simulator for all options.
- Component level:
  JTAG (and JTAG link) probing will be available on all options in the lab and on fields prototypes.
- System level: lab debugging and field validations:
  Same method could be used (TCP/IP access to large internal data with expert application), transfer rate will vary.

Expert mode
detailed observation of large sets of data

Operational mode
Possible with fast TCP/IP using ARM cores



Hardware link to VME
Transport of data
Transparent links between SoC domain
JTAG link

Implementation between Sep 2016-January 2017
=> COULD BE REUSED FOR OTHER PROJECTS

Code reuse between options?

Is the FPGA selection will determine code reusability?

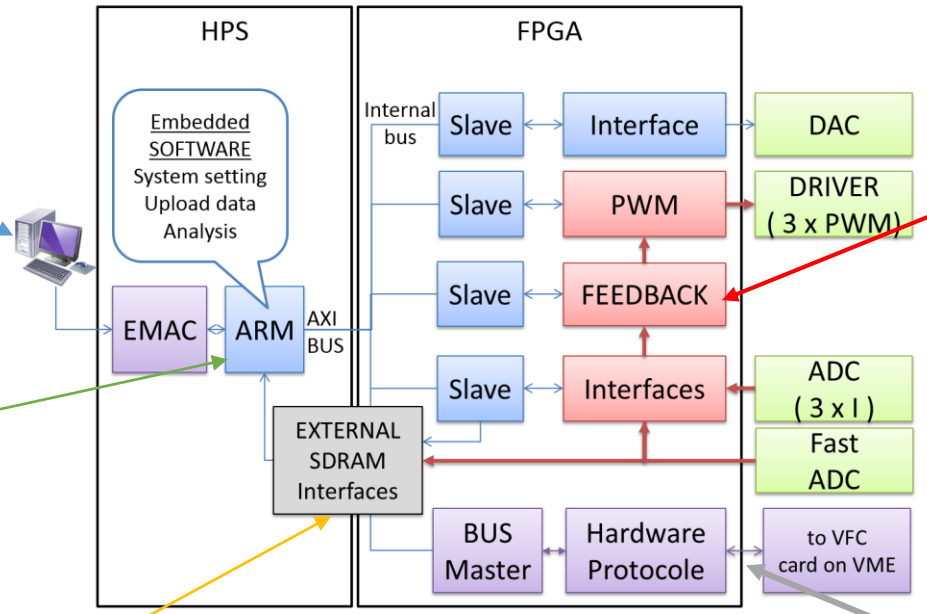
- Yes partially, because we have already large working code (option 2 and 3)
- No, because all main functionalities will be in VHDL (reusable for all options)
- Not really, not much reuse of existing VFC code for all options (no need of VME, BST, etc …)

Expert mode
detailed observation of large sets of data

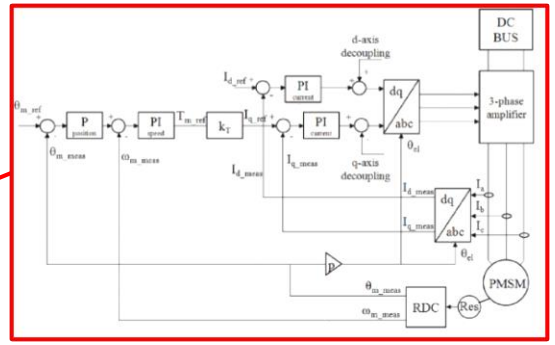
Operational mode
Possible with fast TCP/IP using ARM cores

On-line processing in C
- Monitoring functions
- fast prototyping with access to detailed data

Working version
3 banks (x32) of DDR3
1 bank for ARM cores
1 bank for Fast ADC
1 bank for Feedback data



Ok for 2016
Update in 2017



Hardware link to VME
Transport of data
Transparent links between SoC domain
JTAG link

Implementation between Sep 2016-January 2017
=> COULD BE REUSED FOR OTHER PROJECTS

Design related criteria

Is the FPGA selection will drive design methodology?

- Yes, hardware processors can allow on-line prototype of processing to run in real time.

Methodology:

1) Prototyping data processing in MatLab on existing raw data
2) Simulink modelling of the algorithms and test on Dspace
3) Implementation in C => Fast to write in C and test on real system
   Needs fast processing units
   Needs memory access to data being recorded
4) Final version must be in VHDL:
   - Parallel processing independent to any OS or other running tasks
   - Powerful verification in simulation
   - Powerful tools to do verification on the FPGA
   - But: Long development & verification time

Challenges of the ID processing

- Position, speed and torque precise controlling
  fully written in VHDL
  first version operational for 2016
  Second version foreseen in 2017 (improve precision and flexibility)
- On-line data processing and fault detection
  Will be used for survey system conditions (mechanical, electrical, controls)
  Prototyping foreseen in Simulink/MatLab and C in the drive
  Implemented in VHDL for critical ones, leave
- Special functionalities
  Processing/Area to foresee for future functionalities:
  Tails measurements procedure, Delayed multi-scans (reconstruct small beams), vibrations on-line compensation.

Summary table

- The 3 options where compared for their compatibility with existing hardware, firmware and future needs.
- The table next slide benchmark the 3 options between them.
- This doesn't mean we can’t use one of the 3 to build the scanner.

We will adapt the hardware, firmware and specification based on the selected option by BI management.

Summary table

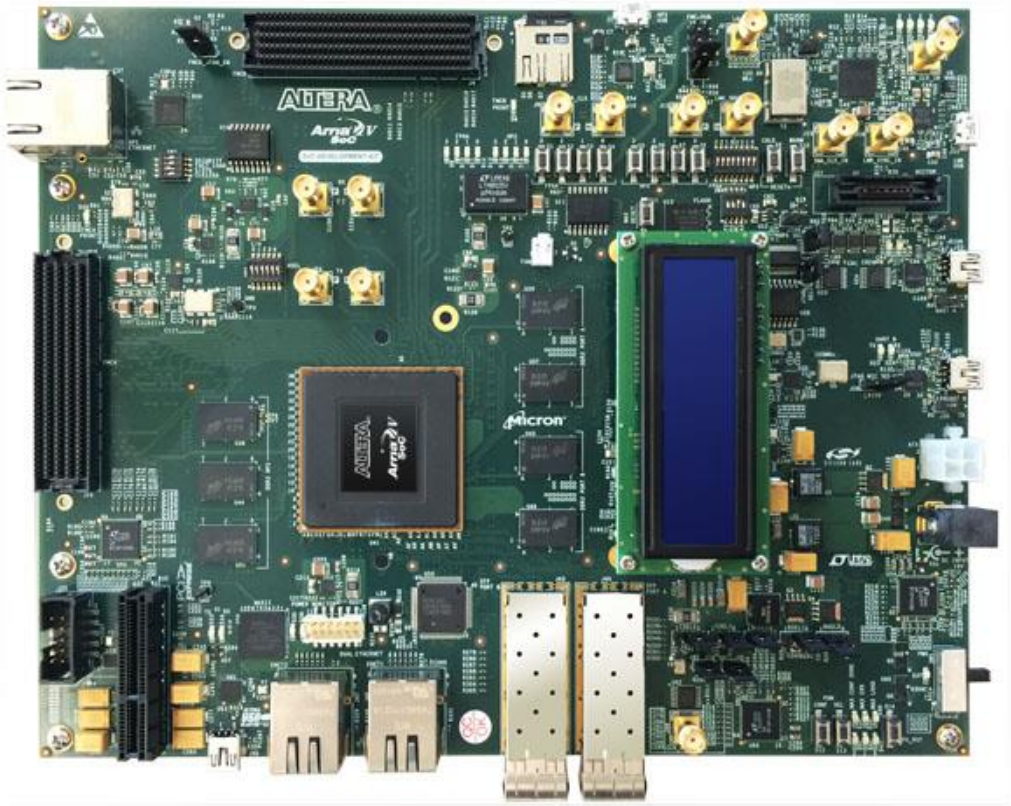
Criteria	Option 1: VFC modified	Option 2: Custom	Option 3: DevKit

Scenarios

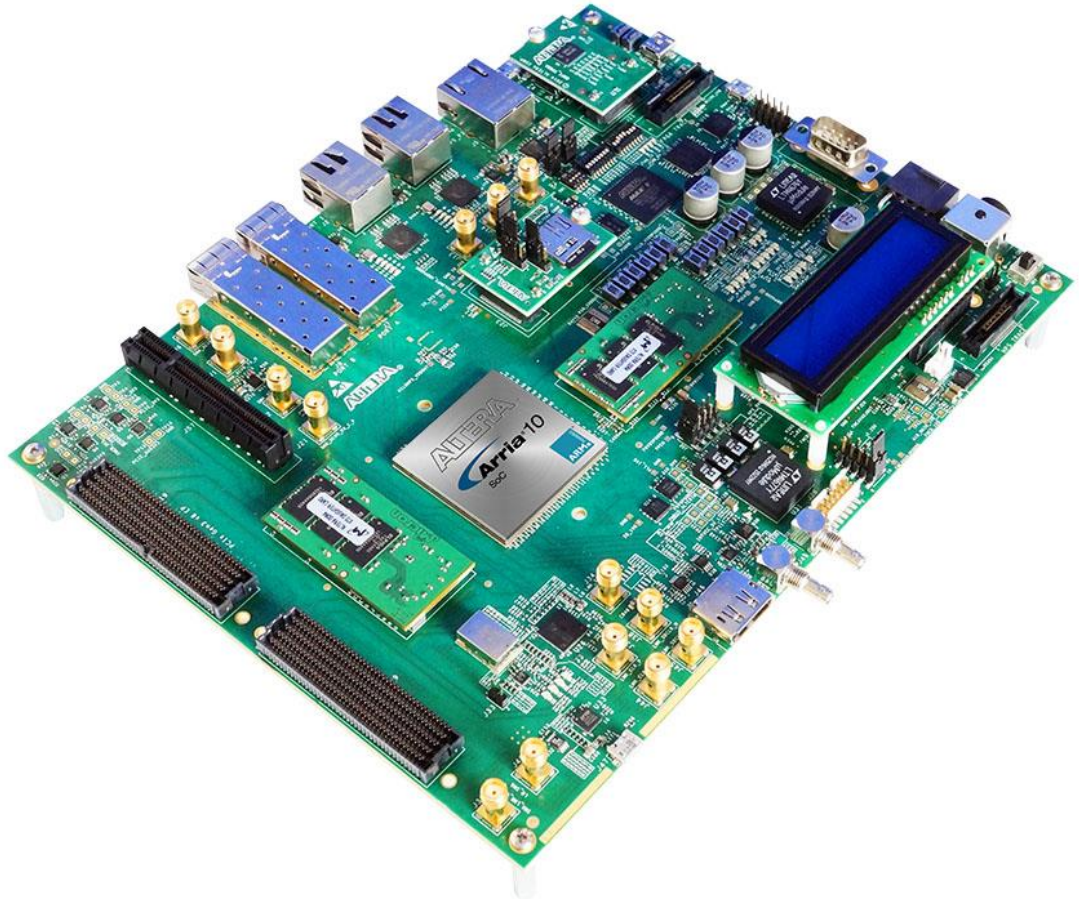
- Possible to run in parallel with different options
- Evaluate VFC, then decide
- ?

Additional slides

Arria SoC kits



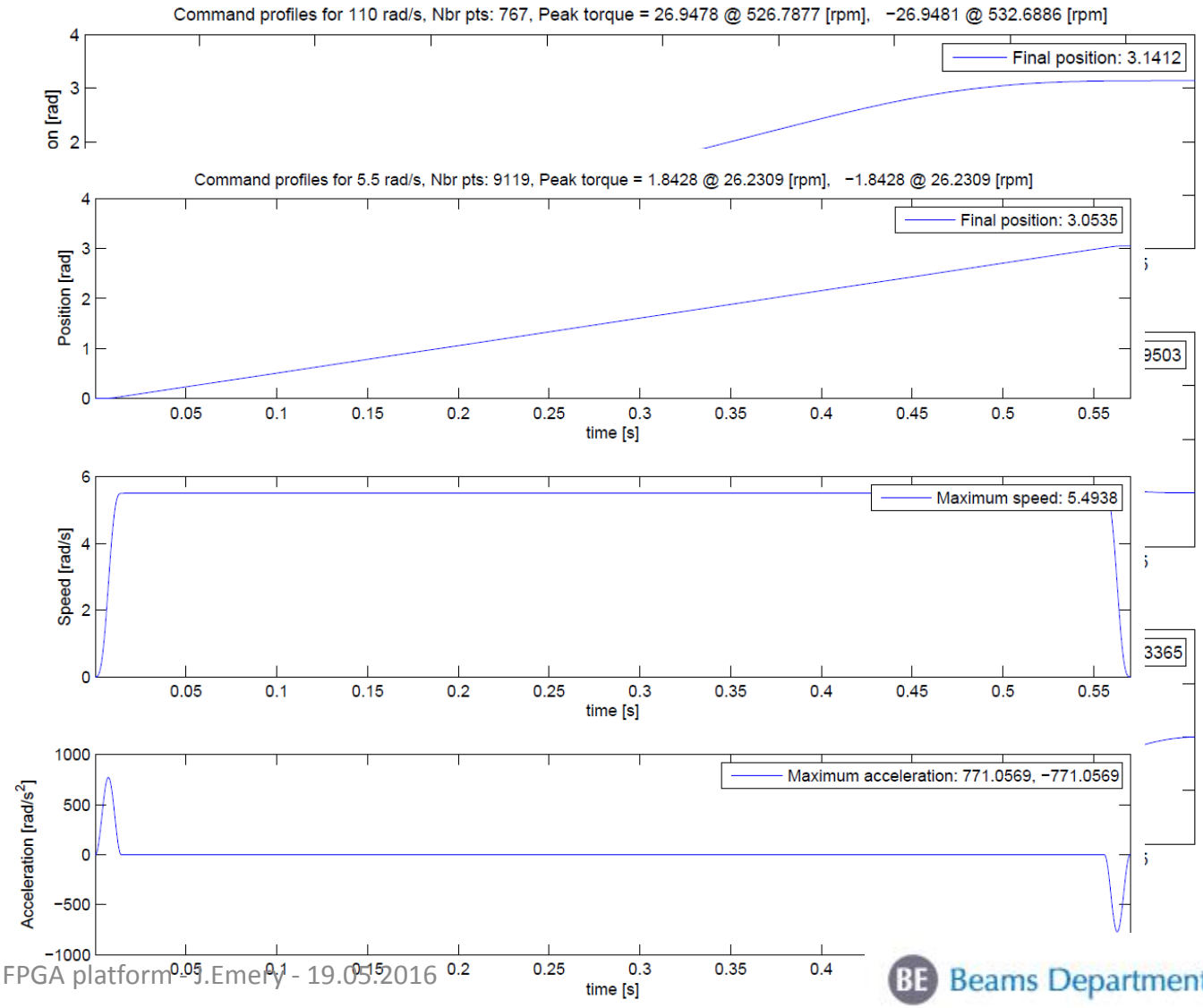
https://www.altera.com/products/boards_and_kits/dev-kits/altera/kit-arria-v-soc.html



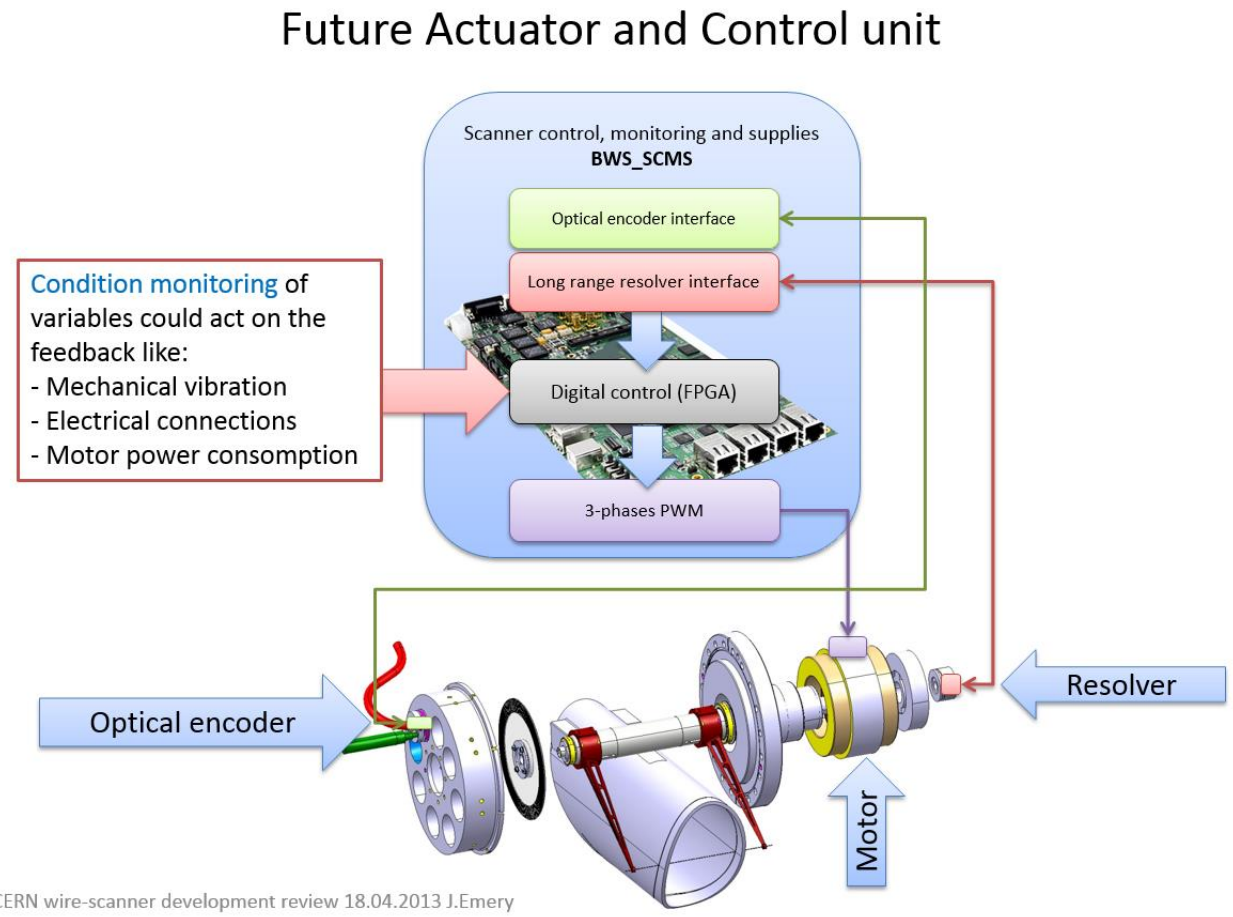
https://www.altera.com/products/boards_and_kits/dev-kits/altera/arria-10-soc-development-kit.html

Profiles: online calculation vs pre-calculated

- Operate at different top speed
  20 [m/s] -> 48 [ms] (767 pts)
  1 [m/s] -> 570 [ms] (9119 pts)
- Today pre-calculated into 3 tables included in the FPGA as ROM (safe).
- Needs 3 tables for each preset
- Alternative:
  Online calculation based on system properties. (3 parameters to play with: Jmax, duration, ratio acc/cst speed).
  Optimised iterative Algorithm to be written in C and monitored by FPGA

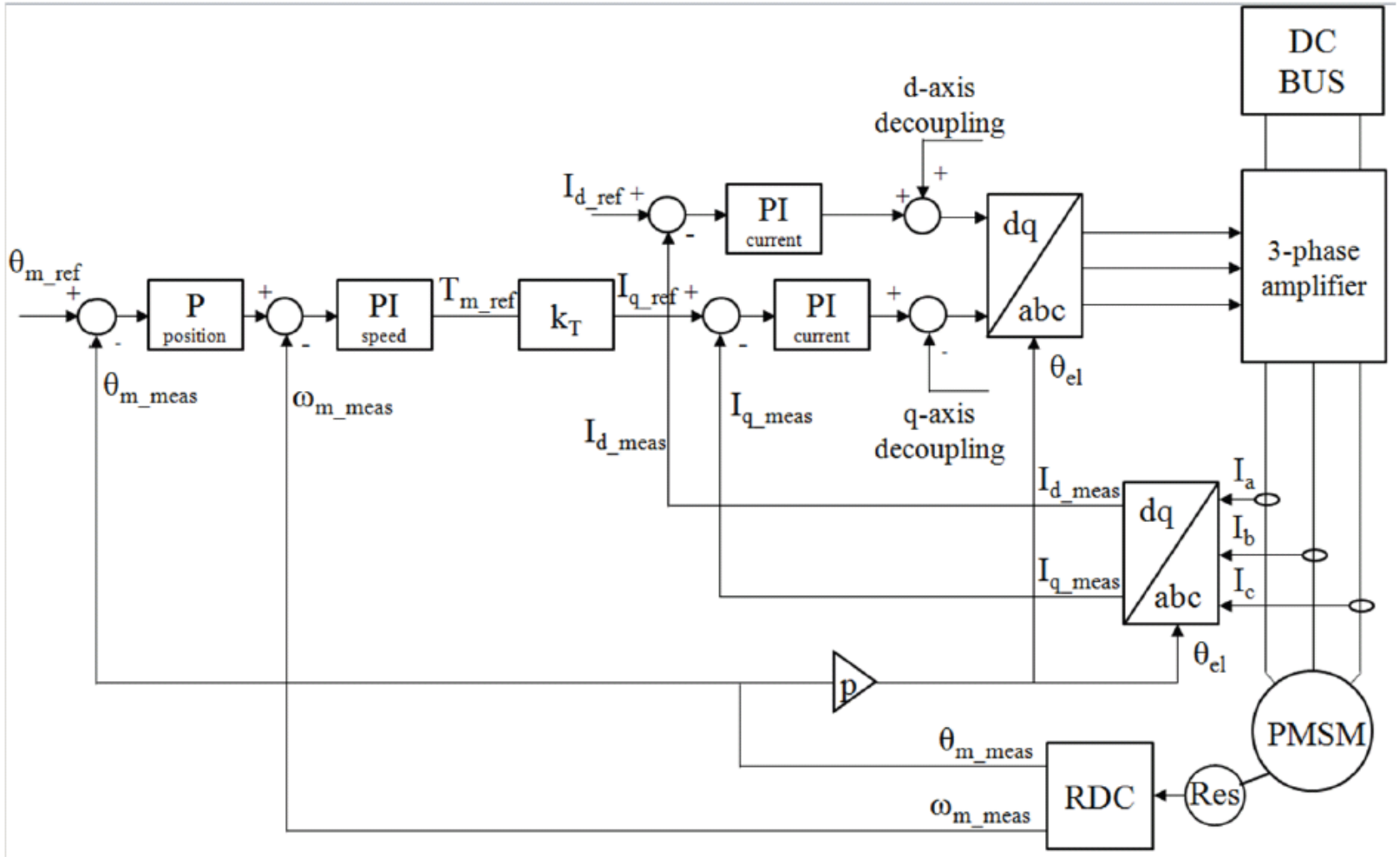


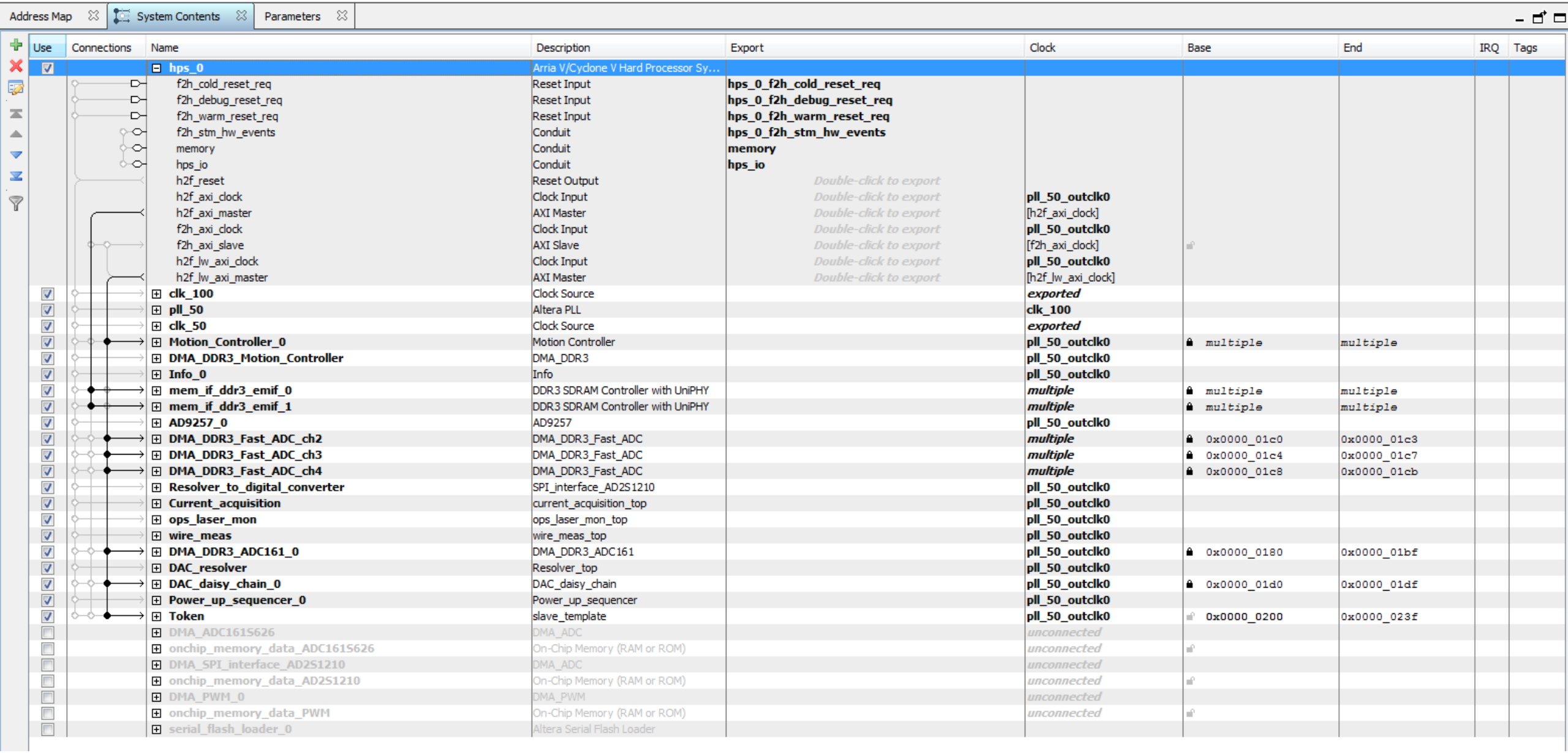
Condition monitoring: Survey all system variables



- Condition monitoring and decision in real time.
- To react to unexpected even during a movement
- Large number of parameters to take into account
- Target reaction time within one feedback period 62.5us:
  12k instructions Nios
  62k instructions ARM

Feedback implementation in VHDL




Address Map
System Contents
Parameters
Use
Connections
Name
Description
Export
Clock
Base
End
IRQ
Tags
hps_0
Arria V/Cyclone V Hard Processor Sy...
f2h_cold_reset_req
Reset Input
hps_0_f2h_cold_reset_req
f2h_debug_reset_req
Reset Input
hps_0_f2h_debug_reset_req
f2h_warm_reset_req
Reset Input
hps_0_f2h_warm_reset_req
f2h_stm_hw_events
Conduit
hps_0_f2h_stm_hw_events
memory
Conduit
memory
hps_io
Conduit
hps_io
h2f_reset
Reset Output
h2f_axi_clock
Clock Input
pll_50_outclk0
h2f_axi_master
AXI Master
[h2f_axi_clock]
f2h_axi_clock
Clock Input
pll_50_outclk0
f2h_axi_slave
AXI Slave
[f2h_axi_clock]
h2f_lw_axi_clock
Clock Input
pll_50_outclk0
h2f_lw_axi_master
AXI Master
[h2f_lw_axi_clock]
Double-click to export
clk_100
Clock Source
exported
pll_50
Altera PLL
clk_100
clk_50
Clock Source
exported
Motion_Controller_0
Motion Controller
pll_50_outclk0
multiple
multiple
DMA_DDR3_Motion_Controller
DMA_DDR3
pll_50_outclk0
Info_0
Info
pll_50_outclk0
mem_if_ddr3_emif_0
DDR3 SDRAM Controller with UniPHY
multiple
multiple
multiple
mem_if_ddr3_emif_1
DDR3 SDRAM Controller with UniPHY
multiple
multiple
multiple
AD9257_0
AD9257
pll_50_outclk0
DMA_DDR3_Fast_ADC_ch2
DMA_DDR3_Fast_ADC
multiple
0x0000_01c0
0x0000_01c3
DMA_DDR3_Fast_ADC_ch3
DMA_DDR3_Fast_ADC
multiple
0x0000_01c4
0x0000_01c7
DMA_DDR3_Fast_ADC_ch4
DMA_DDR3_Fast_ADC
multiple
0x0000_01c8
0x0000_01cb
Resolver_to_digital_converter
SPI_interface_AD2S1210
pll_50_outclk0
Current_acquisition
current_acquisition_top
pll_50_outclk0
ops_laser_mon
ops_laser_mon_top
pll_50_outclk0
wire_meas
wire_meas_top
pll_50_outclk0
DMA_DDR3_ADC161_0
DMA_DDR3_ADC161
pll_50_outclk0
0x0000_0180
0x0000_01bf
DAC_resolver
Resolver_top
pll_50_outclk0
DAC_daisy_chain_0
DAC_daisy_chain
pll_50_outclk0
0x0000_01d0
0x0000_01df
Power_up_sequencer_0
Power_up_sequencer
pll_50_outclk0
Token
slave_template
pll_50_outclk0
0x0000_0200
0x0000_023f
DMA_ADC161S626
DMA_ADC
unconnected
onchip_memory_data_ADC161S626
On-Chip Memory (RAM or ROM)
unconnected
DMA_SPI_interface_AD2S1210
DMA_ADC
unconnected
onchip_memory_data_AD2S1210
On-Chip Memory (RAM or ROM)
unconnected
DMA_PWM_0
DMA_PWM
unconnected
onchip_memory_data_PWM
On-Chip Memory (RAM or ROM)
unconnected
serial_flash_loader_0
Altera Serial Flash Loader

ghrd_5astfd5k3 > mem_if_ddr3_emif_1

DDR3 SDRAM Controller with UniPHY

altera_mem_if_ddr3_emif

Parameters

Generation of the DDR3 Controller with UniPHY produces unencrypted PHY and Controller HDL, constraint scripts, an example design and a testbench for simulation.

Interface Type

☑ Enable Hard External Memory Interface

PHY Settings | Memory Parameters | Memory Timing | Board Settings | Controller Settings | Diagnostics

General Settings

Speed Grade:	3
☐ Generate PHY only	

Clocks

Memory clock frequency:	400.0	MHz
Achieved memory clock frequency:	400.0	MHz
PLL reference clock frequency:	100.0	MHz
Rate on Avalon-MM interface:	Full	
Achieved local clock frequency:	400.0	MHz
☐ Enable AFI half rate clock		

Advanced PHY Settings

☐ Advanced clock phase control		
Additional address and command clock phase:	0.0	Degrees
Supply Voltage:	1.5V DDR3	
I/O standard:	SSTL-15	
PLL sharing mode:	No sharing	
DLL sharing mode:	No sharing	
OCT sharing mode:	Slave	
Reconfigurable PLL Location:	Top_Bottom	

ghrd_5astfd5k3 > mem_if_ddr3_emif_1

DDR3 SDRAM Controller with UniPHY

altera_mem_if_ddr3_emif

Parameters

Generation of the DDR3 Controller with UniPHY produces unencrypted PHY and Controller HDL, constraint scripts, an example design and a testbench for simulation.

Interface Type

☑ Enable Hard External Memory Interface

PHY Settings | Memory Parameters | Memory Timing | Board Settings | Controller Settings | Diagnostics

Apply memory parameters from the manufacturer data sheet
Apply device presets from the preset list on the right.

Memory vendor:	Micron	
Memory format:	Discrete Device	
Memory device speed grade:	800.0	MHz
Total interface width:	32	
DQ/DQS group size:	8	
Number of DQS groups:	4	
Number of chip selects:	1	
Number of clocks:	1	
Row address width:	15	
Column address width:	9	
Bank-address width:	3	
☑ Enable DM pins		

Memory Initialization Options

Mirror Addressing: 1 per chip select:	0
☐ Address and command parity	

Mode Register 0

Read Burst Type:	Sequential
DLL precharge power down:	DLL off
Memory CAS latency setting:	7

Mode Register 1

Output drive strength setting:	RZQ/6
Memory additive CAS latency setting:	Disabled
ODT Rtt nominal value:	RZQ/4

Mode Register 2

Auto selfrefresh method:	Manual
Selfrefresh temperature:	Normal
Memory write CAS latency setting:	6
Dynamic ODT (Rtt_WR) value:	Dynamic ODT off

ghrd_5astfd5k3 > mem_if_ddr3_emif_1

DDR3 SDRAM Controller with UniPHY

altera_mem_if_ddr3_emif

Parameters

Generation of the DDR3 Controller with UniPHY produces unencryp
constraint scripts, an example design and a testbench for simulatio

Interface Type

☑ Enable Hard External Memory Interface

PHY Settings | Memory Parameters | Memory Timing | Board Sett

Apply timing parameters from the manufacturer data sheet
Apply device presets from the preset list on the right.

tIS (base):	170	ps
tIH (base):	120	ps
tDS (base):	10	ps
tDH (base):	45	ps
tDQSQ:	100	ps
tQH:	0.38	cycles
tDQSCK:	255	ps
tDQSS:	0.27	cycles
tQSH:	0.4	cycles
tDSH:	0.18	cycles
tDSS:	0.18	cycles
tINIT:	500	us
tMRD:	4	cycles
tRAS:	35.0	ns
tRCD:	13.75	ns
tRP:	13.75	ns
tREFI:	7.8	us
tRFC:	260.0	ns
tWR:	15.0	ns
tWTR:	4	cycles
tFAW:	35.0	ns
tRRD:	6.0	ns
tRTP:	7.5	ns

ghrd_5astfd5k3 > mem_if_ddr3_emif_1

DDR3 SDRAM Controller with UniPHY

altera_mem_if_ddr3_emif

Parameters

Generation of the DDR3 Controller with UniPHY produces unencrypted PHY and Controller HDL, constraint scripts, an example design and a testbench for simulation.

Interface Type

☑ Enable Hard External Memory Interface

PHY Settings | Memory Parameters | Memory Timing | Board Settings | Controller Settings | Diagnostics

Use the Board Settings to model the board-level effects in the timing analysis.

The wizard supports single- and multi-rank configurations. Altera has determined the effects on the output signaling of these configurations and has stored the effects on the output slew rate and the channel uncertainty within the UniPHY MegaWizard.

These values are representative of specific Altera boards. You must change the values to account for the board level effects for your board. You can use HyperLynx or similar simulators to obtain values that are representative of your board.

Setup and Hold Derating

You can specify the slew rate of the output signals to refer to their effect on the setup and hold times of both the address and command signals and the DQ signals, or specify the setup and hold times directly.

Derating method:

- (•) Use Altera's default settings
- ( ) Specify slew rates to calculate setup and hold times
- ( ) Specify setup and hold times directly

CK/CK# slew rate (Differential):	2.0	V/ns
Address and command slew rate:	1.0	V/ns
DQS/DQS# slew rate (Differential):	2.0	V/ns
DQ slew rate:	1.0	V/ns
tIS:	0.32	ns
tIH:	0.22	ns
tDS:	0.16	ns
tDH:	0.145	ns

Channel Signal Integrity

Channel Signal Integrity is a measure of the distortion of the eye due to intersymbol interference or crosstalk or other effects. Typically when going from a single-rank configuration to a multi-rank configuration there is an increase in the channel loss as there are multiple stubs causing reflections. Please perform your channel signal integrity simulations and enter the extra channel uncertainty as compared to Altera's reference eye diagram.

Derating Method:

- (•) Use Altera's default settings
- ( ) Specify channel uncertainty values

Address and command eye reduction (setup):	0.0	ns
Address and command eye reduction (hold):	0.0	ns
Write DQ eye reduction:	0.0	ns
Write Delta DQS arrival time:	0.0	ns
Read DQ eye reduction:	0.0	ns
Read Delta DQS arrival time:	0.0	ns

Board Skews

PCB traces can have skews between them that can cause timing margins to be reduced. Furthermore skews between different ranks can further reduce the timing margin in multi-rank topologies.

Restore default values

Maximum CK delay to DIMM/device:	0.6	ns
Maximum DQS delay to DIMM/device:	0.6	ns
Minimum delay difference between CK and DQS:	-0.01	ns
Maximum delay difference between CK and DQS:	0.01	ns
Maximum skew within DQS group:	0.02	ns
Maximum skew between DQS groups:	0.02	ns
Average delay difference between DQ and DQS:	0.0	ns
Maximum skew within address and command bus:	0.02	ns
Average delay difference between address and command and CK:	0.0	ns

ghrd_5astfd5k3 > mem_if_ddr3_emif_1

DDR3 SDRAM Controller with UniPHY

altera_mem_if_ddr3_emif

Parameters

Generation of the DDR3 Controller with UniPHY produces unencrypted PHY and Controller HDL, constraint scripts, an example design and a testbench for simulation.

Interface Type

☑ Enable Hard External Memory Interface

PHY Settings | Memory Parameters | Memory Timing | Board Settings | Controller Settings | Diagnostics

Avalon Interface

☑ Generate power-of-2 data bus widths for Qsys or SOPC Builder

☐ Generate SOPC Builder compatible resets

Maximum Avalon-MM burst length: 4

☐ Enable Avalon-MM byte-enable signal

Low Power Mode

☐ Enable Self-Refresh Controls

☐ Enable Auto Power-Down

Auto Power-Down Cycles: 0 cycles

Efficiency

☐ Enable User Auto-Refresh Controls

☐ Enable Auto-Precharge Control

Local-to-Memory Address Mapping: CHIP-ROW-BANK-COL

Command Queue Look-Ahead Depth: 4

☑ Enable Reordering

Starvation limit for each command: 10 commands

Configuration, Status and Error Handling

☐ Enable Configuration and Status Register Interface

CSR port host interface: Internal (JTAG)

☐ Enable Error Detection and Correction Logic

☐ Enable Auto Error Correction

Multiple Port Front End

☐ Export bonding interface

☐ Expand Avalon-MM data for ECC

Number of ports: 4

Port	Type	Width	Priority	Weight	Allocated Write-Data FIFO	Allocated Read-Data FIFO
Port 0	**Read-only**	**32**	**1**	**0**	None	F0
Port 1	**Write-only**	**32**	**2**	**0**	F0	None
Port 2	**Write-only**	**32**	**2**	**0**	F1	None
Port 3	**Write-only**	**32**	**2**	**0**	F2	None

ghrd_5astfd5k3 > mem_if_ddr3_emif_1

DDR3 SDRAM Controller with UniPHY

altera_mem_if_ddr3_emif

Parameters

Generation of the DDR3 Controller with UniPHY produces unencrypted PHY and Controller HDL, constraint scripts, an example design and a testbench for simulation.

Interface Type

☑ Enable Hard External Memory Interface

PHY Settings | Memory Parameters | Memory Timing | Board Settings | Controller Settings | Diagnostics

Simulation Options

Auto-calibration mode: Full calibration

☑ Skip Memory Initialization Delays

☑ Enable verbose memory model output

☐ Enable support for Nios II ModelSim flow in Eclipse

Debugging Options

Debugging feature set: Option 1

Feature Set	Included Debugging Features	Additional Utilization
No Debugging	None	None
Option 1	Connectivity to the EMIF toolkit allowing you to display information about your interface and generate reports.	+600 Registers +700 ALUTs +8 M9Ks

☐ Enable EMIF On-Chip Debug Toolkit

EMIF On-Chip Debug Toolkit interface type: Avalon-MM Slave

Efficiency Monitor and Protocol Checker Settings

The Efficiency Monitor and Protocol Checker is used to measure efficiency on the Avalon interface between the Traffic Generator and the Controller. It will also perform protocol checking on the bus.

☐ Enable the Efficiency Monitor and Protocol Checker on the Controller Avalon Interface